5 Random Walks and Markov Chains

A random walk on a directed graph consists of a sequence of vertices generated from a start vertex by selecting an edge, traversing the edge to a new vertex, and repeating the process. We will see that if the graph is strongly connected, then the fraction of time the walk spends at the various vertices of the graph converges to a stationary probability distribution.

Since the graph is directed, there might be vertices with no out edges and hence nowhere for the walk to go. Vertices in a strongly connected component with no in edges from the remainder of the graph can never be reached unless the component contains the start vertex. Once a walk leaves a strongly connected component it can never return. Most of our discussion of random walks will involve strongly connected graphs.

Start a random walk at a vertex x_0 and think of the starting probability distribution as putting a mass of one on x_0 and zero on every other vertex. More generally, one could start with any probability distribution $\mathbf{p}$, where $\mathbf{p}$ is a row vector with nonnegative components summing to one, with p_x being the probability of starting at vertex x. The probability of being at vertex x at time $t+1$ is the sum over each adjacent vertex y of being at y at time t and taking the transition from y to x. Let $\mathbf{p^{(t)}}$ be a row vector with a component for each vertex specifying the probability mass of the vertex at time t and let $\mathbf{p^{(t+1)}}$ be the row vector of probabilities at time $t+1$. In matrix notation[4]

$$\mathbf{p^{(t)}}P = \mathbf{p^{(t+1)}}$$

where the ij^{th} entry of the matrix P is the probability of the walk at vertex i selecting the edge to vertex j.

A fundamental property of a random walk is that in the limit, the long-term average probability of being at a particular vertex is independent of the start vertex, or an initial probability distribution over vertices, provided only that the underlying graph is strongly connected. The limiting probabilities are called the *stationary probabilities*. This fundamental theorem is proved in the next section.

A special case of random walks, namely random walks on undirected graphs, has important connections to electrical networks. Here, each edge has a parameter called *conductance*, like the electrical conductance, and if the walk is at vertex u, it chooses the edge from among all edges incident to u to walk to the next vertex with probabilities proportional to their conductance. Certain basic quantities associated with random walks are hitting time, the expected time to reach vertex y starting at vertex x, and cover time, the expected time to visit every vertex. Qualitatively, these quantities are all bounded above by polynomials in the number of vertices. The proofs of these facts will rely on the

[4]Probability vectors are represented by row vectors to simplify notation in equations like the one here.

random walk	**Markov chain**
graph	stochastic process
vertex	state
strongly connected	persistent
aperiodic	aperiodic
strongly connected and aperiodic	ergotic
undirected graph	time reversible

Table 5.1: Correspondence between terminology of random walks and Markov chains

analogy between random walks and electrical networks.

Aspects of the theory of random walks was developed in computer science with an important application in defining the pagerank of pages on the World Wide Web by their stationary probability. An equivalent concept called a *Markov chain* had previously been developed in the statistical literature. A Markov chain has a finite set of *states*. For each pair of states x and y, there is a *transition probability* p_{xy} of going from state x to state y where for each x, $\sum_y p_{xy} = 1$. A random walk in the Markov chain starts at some state. At a given time step, if it is in state x, the next state y is selected randomly with probability p_{xy}. A Markov chain can be represented by a directed graph with a vertex representing each state and an edge with weight p_{xy} from vertex x to vertex y. We say that the Markov chain is *connected* if the underlying directed graph is strongly connected. That is, if there is a directed path from every vertex to every other vertex. The matrix P consisting of the p_{xy} is called the *transition probability matrix* of the chain. The terms "random walk" and "Markov chain" are used interchangeably. The correspondence between the terminologies of random walks and Markov chains is given in Table 5.1.

A state of a Markov chain is *persistent* if it has the property that should the state ever be reached, the random process will return to it with probability one. This is equivalent to the property that the state is in a strongly connected component with no out edges. For most of the chapter, we assume that the underlying directed graph is strongly connected. We discuss here briefly what might happen if we do not have strong connectivity. Consider the directed graph in Figure 5.1b with three strongly connected components, A, B, and C. Starting from any vertex in A, there is a nonzero probability of eventually reaching any vertex in A. However, the probability of returning to a vertex in A is less than one and thus vertices in A, and similarly vertices in B, are not persistent. From any vertex in C, the walk eventually will return with probability one to the vertex, since there is no way of leaving component C. Thus, vertices in C are persistent.

Markov chains are used to model situations where all the information of the system

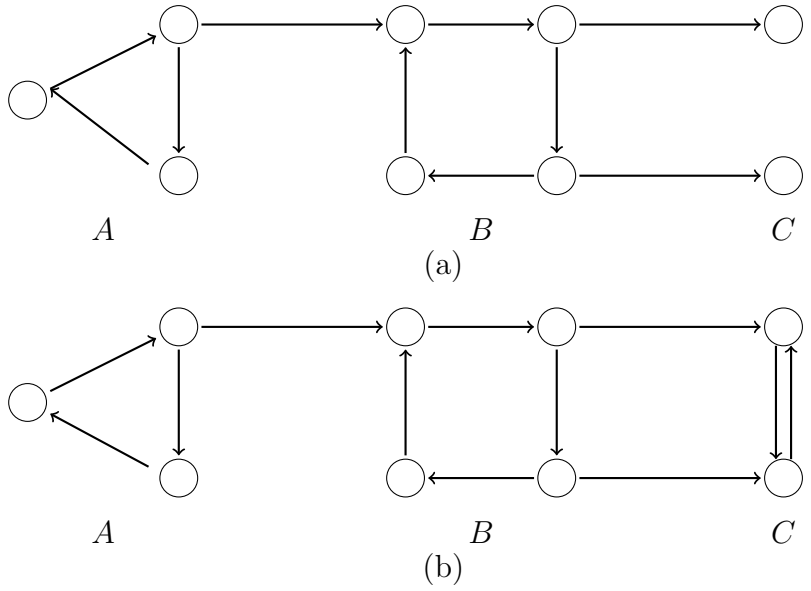


Figure 5.1: (a) A directed graph with vertices having no out out edges and a strongly connected component A with no in edges.
(b) A directed graph with three strongly connected components.

necessary to predict the future can be encoded in the current state. A typical example is speech, where for a small k the current state encodes the last k syllables uttered by the speaker. Given the current state, there is a certain probability of each syllable being uttered next and these can be used to calculate the transition probabilities. Another example is a gambler's assets, which can be modeled as a Markov chain where the current state is the amount of money the gambler has on hand. The model would only be valid if the gambler's bets depend only on current assets, not the past.

Later in the chapter, we study the widely used Markov Chain Monte Carlo method (MCMC). Here, the objective is to sample a large space according to some probability distribution p. The number of elements in the space may be very large, say 10^{100}. One designs a Markov chain where states correspond to the elements of the space. The transition probabilities of the chain are designed so that the stationary probability of the chain is the probability distribution p with which we want to sample. One samples by taking a random walk until the probability distribution is close to the stationary distribution of the chain and then selects the point the walk is at. The walk continues a number of steps until the probability distribution is no longer dependent on where the walk was when the first element was selected. A second point is then selected, and so on. Although it is impossible to store the graph in a computer since it has 10^{100} vertices, to do the walk one needs only store the vertex the walk is at and be able to generate the adjacent vertices by some algorithm. What is critical is that the probability of the walk converges to the stationary probability in time logarithmic in the number of states.

We mention two motivating examples. The first is to estimate the probability of a region R in d-space according to a probability density like the Gaussian. Put down a grid and make each grid point that is in R a state of the Markov chain. Given a probability density p, design transition probabilities of a Markov chain so that the stationary distribution is exactly p. In general, the number of states grows exponentially in the dimension d, but the time to converge to the stationary distribution grows polynomially in d.

A second example is from physics. Consider an $n \times n$ grid in the plane with a particle at each grid point. Each particle has a spin of ± 1. There are 2^{n^2} spin configurations. The energy of a configuration is a function of the spins. A central problem in statistical mechanics is to sample a spin configuration according to their probability. It is easy to design a Markov chain with one state per spin configuration so that the stationary probability of a state is proportional to the state's energy. If a random walk gets close to the stationary probability in time polynomial to n rather than 2^{n^2}, then one can sample spin configurations according to their probability.

A quantity called the *mixing time*, loosely defined as the time needed to get close to the stationary distribution, is often much smaller than the number of states. In Section 5.8, we relate the mixing time to a combinatorial notion called *normalized conductance* and derive good upper bounds on the mixing time in many cases.

5.1 Stationary Distribution

Let $\mathbf{p^{(t)}}$ be the probability distribution after t steps of a random walk. Define the *long-term probability distribution* $\mathbf{a^{(t)}}$ by

$$\mathbf{a^{(t)}} = \frac{1}{t}\left(\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}}\right).$$

The fundamental theorem of Markov chains asserts that the long-term probability distribution of a connected Markov chain converges to a unique limit probability vector, which we denote by $\boldsymbol{\pi}$. Executing one more step, starting from this limit distribution, we get back the same distribution. In matrix notation, $\boldsymbol{\pi}P = \boldsymbol{\pi}$ where P is the matrix of transition probabilities. In fact, there is a unique probability vector (nonnegative components summing to one) satisfying $\boldsymbol{\pi}P = \boldsymbol{\pi}$ and this vector is the limit. Also since one step does not change the distribution, any number of steps would not either. For this reason, $\boldsymbol{\pi}$ is called the *stationary distribution*.

Before proving the fundamental theorem of Markov chains, we first prove a technical lemma.

Lemma 5.1 *Let P be the transition probability matrix for a connected Markov chain. The $n \times (n+1)$ matrix $A = [P - I\ ,\ \mathbf{1}]$ obtained by augmenting the matrix $P - I$ with an additional column of ones has rank n.*

Proof: If the rank of $A = [P - I, \mathbf{1}]$ was less than n there would be two linearly independent solutions to $A\mathbf{x} = \mathbf{0}$. Each row in P sums to one so each row in $P - I$ sums to zero. Thus $\mathbf{x} = (\mathbf{1}, 0)$, where all but the last coordinate of $\mathbf{x}$ is 1, is one solution to $A\mathbf{x} = \mathbf{0}$. Assume there was a second solution $(\mathbf{x}, \alpha)$ perpendicular to $(\mathbf{1}, 0)$. Then $(P-I)\mathbf{x}+\alpha\mathbf{1} = \mathbf{0}$ or $x_i = \sum_j p_{ij}x_j + \alpha$. Each x_i is a convex combination of some x_j plus α. Let S be the set of i for which x_i attains its maximum value. $\bar{S}$ is not empty since x is perpendicular to $\mathbf{1}$ and hence $\sum_j x_j = 0$. Connectedness implies that some x_k of maximum value is adjacent to some x_l of lower value. Thus, $x_k > \sum_j p_{kj}x_j$. Therefore α must be greater than 0 in $x_k = \sum_j p_{kj}x_j + \alpha$..

A symmetric argument with T the set of i with x_i taking its minimum value implies $\alpha < 0$ producing a contradiction thereby proving the lemma. ∎

Theorem 5.2 (Fundamental Theorem of Markov Chains) *For a connected Markov chain there is a unique probability vector $\boldsymbol{\pi}$ satisfying $\boldsymbol{\pi}P = \boldsymbol{\pi}$. Moreover, for any starting distribution, $\lim_{t\to\infty} \mathbf{a}^{(\mathbf{t})}$ exists and equals $\boldsymbol{\pi}$.*

Proof: Note that $\mathbf{a}^{(\mathbf{t})}$ is itself a probability vector, since its components are nonnegative and sum to 1. Run one step of the Markov chain starting with distribution $\mathbf{a}^{(\mathbf{t})}$; the distribution after the step is $\mathbf{a}^{(\mathbf{t})}P$. Calculate the change in probabilities due to this step.

$$\begin{aligned}
\mathbf{a}^{(\mathbf{t})}P - \mathbf{a}^{(\mathbf{t})} &= \frac{1}{t}\left[\mathbf{p}^{(\mathbf{0})}P + \mathbf{p}^{(\mathbf{1})}P + \cdots + \mathbf{p}^{(\mathbf{t-1})}P\right] - \frac{1}{t}\left[\mathbf{p}^{(\mathbf{0})} + \mathbf{p}^{(\mathbf{1})} + \cdots + \mathbf{p}^{(\mathbf{t-1})}\right] \\
&= \frac{1}{t}\left[\mathbf{p}^{(\mathbf{1})} + \mathbf{p}^{(\mathbf{2})} + \cdots + \mathbf{p}^{(\mathbf{t})}\right] - \frac{1}{t}\left[\mathbf{p}^{(\mathbf{0})} + \mathbf{p}^{(\mathbf{1})} + \cdots + \mathbf{p}^{(\mathbf{t-1})}\right] \\
&= \frac{1}{t}\left(\mathbf{p}^{(\mathbf{t})} - \mathbf{p}^{(\mathbf{0})}\right).
\end{aligned}$$

Thus, $\mathbf{b}^{(\mathbf{t})} = \mathbf{a}^{(\mathbf{t})}P - \mathbf{a}^{(\mathbf{t})}$ satisfies $|\mathbf{b}^{(\mathbf{t})}| \leq \frac{2}{t} \to 0$, as $t \to \infty$.

By Lemma 5.1 above, A has rank n. The $n \times n$ submatrix B of A consisting of all its columns except the first is invertible. Let $\mathbf{c}^{(\mathbf{t})}$ be obtained from $\mathbf{b}^{(\mathbf{t})}$ by removing the first entry. Then, $\mathbf{a}^{(\mathbf{t})}B = [\mathbf{c}^{(\mathbf{t})}, 1]$ and so $\mathbf{a}^{(\mathbf{t})} = [\mathbf{c}^{(\mathbf{t})}\ ,\ 1]B^{-1} \to [\mathbf{0}\ ,\ 1]B^{-1}$. We have the theorem with $\boldsymbol{\pi} = [\mathbf{0}\ ,\ 1]B^{-1}$. ∎

Observe that the expected time r_x for a Markov chain starting in state x to return to state x is the reciprocal of the stationary probability of x. That is $r_x = \frac{1}{\pi_x}$. Intuitively this follows by observing that if a long walk always returns to state x in exactly r_x steps, the frequency of being in a state x would be $\frac{1}{r_x}$. A rigorous proof requires the Strong Law of Large Numbers.

We finish this section with the following lemma useful in establishing that a probability distribution is the stationary probability distribution for a random walk on a connected graph with edge probabilities.

Lemma 5.3 *For a random walk on a strongly connected graph with probabilities on the edges, if the vector* $\boldsymbol{\pi}$ *satisfies* $\pi_x p_{xy} = \pi_y p_{yx}$ *for all* x *and* y *and* $\sum_x \pi_x = 1$*, then* $\boldsymbol{\pi}$ *is the stationary distribution of the walk.*

Proof: Since $\boldsymbol{\pi}$ satisfies $\pi_x p_{xy} = \pi_y p_{yx}$, take the sum of both sides to get $\pi_x = \sum_y \pi_y p_{yx}$ and hence $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi} = \boldsymbol{\pi} P$. By Theorem 5.2, $\boldsymbol{\pi}$ is the unique stationary probability. ∎

5.2 Electrical Networks and Random Walks

In the next few sections, we study the relationship between electrical networks and random walks on undirected graphs. The graphs have nonnegative weights on each edge. A step is executed by picking a random edge from the current vertex with probability proportional to the edge's weight and traversing the edge.

An electrical network is a connected, undirected graph in which each edge (x, y) has a resistance $r_{xy} > 0$. In what follows, it is easier to deal with conductance defined as the reciprocal of resistance, $c_{xy} = \frac{1}{r_{xy}}$, rather than resistance. Associated with an electrical network is a random walk on the underlying graph defined by assigning a probability $p_{xy} = \frac{c_{xy}}{c_x}$ to the edge (x, y) incident to the vertex x, where the normalizing constant c_x equals $\sum_y c_{xy}$. Note that although c_{xy} equals c_{yx}, the probabilities p_{xy} and p_{yx} may not be equal due to the normalization required to make the probabilities at each vertex sum to one. We shall soon see that there is a relationship between current flowing in an electrical network and a random walk on the underlying graph.

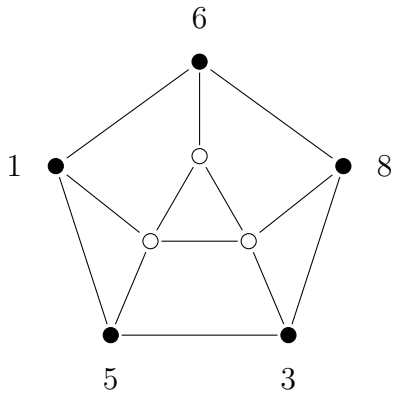
Since we assume that the undirected graph is connected, by Theorem 5.2 there is a unique stationary probability distribution.The stationary probability distribution is $\boldsymbol{\pi}$ where $\pi_x = \frac{c_x}{c_0}$ where $c_0 = \sum_x c_x$. To see this, for all x and y

$$\pi_x p_{xy} = \frac{c_x}{c_0}\frac{c_{xy}}{c_x} = \frac{c_{xy}}{c_0} = \frac{c_y}{c_0}\frac{c_{yx}}{c_y} = \pi_y p_{yx}$$

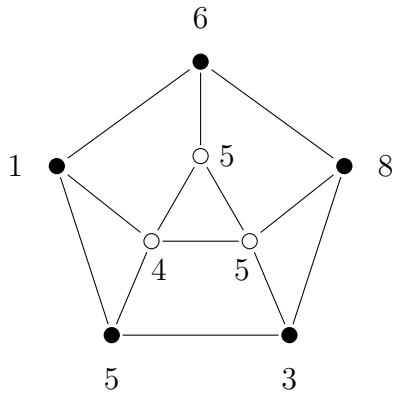
and hence by Lemma 5.3, $\boldsymbol{\pi}$ is the unique stationary probability.

Harmonic functions

Harmonic functions are useful in developing the relationship between electrical networks and random walks on undirected graphs. Given an undirected graph, designate a nonempty set of vertices as boundary vertices and the remaining vertices as interior vertices. A harmonic function g on the vertices is one in which the value of the function at the boundary vertices is fixed to some boundary condition and the value of g at any interior vertex x is a weighted average of the values at all the adjacent vertices y, with



Graph with boundary vertices dark and boundary conditions specified.



Values of harmonic function satisfying boundary conditions

Figure 5.2: Graph illustrating an harmonic function.

weights p_{xy} satisfying $\sum_y p_{xy} = 1$ for each x. Thus, if at every interior vertex x for some set of weights p_{xy} satisfying $\sum_y p_{xy} = 1$, $g_x = \sum_y g_y p_{xy}$, then g is an harmonic function.

Example: Convert an electrical network with conductances c_{xy} to a weighted, undirected graph with probabilities p_{xy}. Let $\mathbf{f}$ be a function satisfying $\mathbf{f}P = \mathbf{f}$ where P is the matrix of probabilities. It follows that the function $g_x = \frac{f_x}{c_x}$ is harmonic.

$$g_x = \frac{f_x}{c_x} = \frac{1}{c_x}\sum_y f_y p_{yx} = \frac{1}{c_x}\sum_y f_y \frac{c_{yx}}{c_y}$$
$$= \frac{1}{c_x}\sum_y f_y \frac{c_{xy}}{c_y} = \sum_y \frac{f_y}{c_y}\frac{c_{xy}}{c_x} = \sum_y g_y p_{xy}$$

∎

A harmonic function on a connected graph takes on its maximum and minimum on the boundary. Suppose the maximum does not occur on the boundary. Let S be the set of interior vertices at which the maximum value is attained. Since S contains no boundary vertices, $\bar{S}$ is nonempty. Connectedness implies that there is at least one edge (x, y) with $x \in S$ and $y \in \bar{S}$. The value of the function at x is the average of the value at its neighbors, all of which are less than or equal to the value at x and the value at y is strictly less, a contradiction. The proof for the minimum value is identical.

There is at most one harmonic function satisfying a given set of equations and boundary conditions. For suppose there were two solutions, $f(x)$ and $g(x)$. The difference of two solutions is itself harmonic. Since $h(x) = f(x) - g(x)$ is harmonic and has value zero on the boundary, by the min and max principles it has value zero everywhere. Thus $f(x) = g(x)$.

The analogy between electrical networks and random walks

There are important connections between electrical networks and random walks on undirected graphs. Choose two vertices a and b. For reference purposes let the voltage v_b equal zero. Attach a current source between a and b so that the voltage v_a equals one. Fixing the voltages at v_a and v_b induces voltages at all other vertices along with a current flow through the edges of the network. The analogy between electrical networks and random walks is the following. Having fixed the voltages at the vertices a and b, the voltage at an arbitrary vertex x equals the probability of a random walk starting at x reaching a before reaching b. If the voltage v_a is adjusted so that the current flowing into vertex a corresponds to one walk, then the current flowing through an edge is the net frequency with which a random walk from a to b traverses the edge.

Probabilistic interpretation of voltages

Before showing that the voltage at an arbitrary vertex x equals the probability of a random walk starting at x reaching a before reaching b, we first show that the voltages form a harmonic function. Let x and y be adjacent vertices and let i_{xy} be the current flowing through the edge from x to y. By Ohm's law,

$$i_{xy} = \frac{v_x - v_y}{r_{xy}} = (v_x - v_y)c_{xy}.$$

By Kirchhoff's law the currents flowing out of each vertex sum to zero.

$$\sum_y i_{xy} = 0$$

Replacing currents in the above sum by the voltage difference times the conductance yields

$$\sum_y (v_x - v_y)c_{xy} = 0$$

or

$$v_x \sum_y c_{xy} = \sum_y v_y c_{xy}.$$

Observing that $\sum_y c_{xy} = c_x$ and that $p_{xy} = \frac{c_{xy}}{c_x}$, yields $v_x c_x = \sum_y v_y p_{xy} c_x$. Hence, $v_x = \sum_y v_y p_{xy}$. Thus, the voltage at each vertex x is a weighted average of the voltages at the adjacent vertices. Hence the voltages form a harmonic function with $\{a, b\}$ as the boundary.

Let p_x be the probability that a random walk starting at vertex x reaches a before b. Clearly $p_a = 1$ and $p_b = 0$. Since $v_a = 1$ and $v_b = 0$, it follows that $p_a = v_a$ and $p_b = v_b$.

Furthermore, the probability of the walk reaching a from x before reaching b is the sum over all y adjacent to x of the probability of the walk going from x to y in the first step and then reaching a from y before reaching b. That is

$$p_x = \sum_y p_{xy} p_y.$$

Hence, p_x is the same harmonic function as the voltage function v_x and $\mathbf{v}$ and $\mathbf{p}$ satisfy the same boundary conditions at a and b.. Thus, they are identical functions. The probability of a walk starting at x reaching a before reaching b is the voltage v_x.

Probabilistic interpretation of current

In a moment, we will set the current into the network at a to have a value which we will equate with one random walk. We will then show that the current i_{xy} is the net frequency with which a random walk from a to b goes through the edge xy before reaching b. Let u_x be the expected number of visits to vertex x on a walk from a to b before reaching b. Clearly $u_b = 0$. Every time the walk visits x, x not equal to a, it must come to x from some vertex y. Thus, the number of visits to x before reaching b is the sum over all y of the number of visits u_y to y before reaching b times the probability p_{yx} of going from y to x. For x not equal to b or a

$$u_x = \sum_{y \neq b} u_y p_{yx}.$$

Since $u_b = 0$ and $c_x p_{xy} = c_y p_{yx}$

$$u_x = \sum_{\text{all } y} u_y \frac{c_x p_{xy}}{c_y}$$

and hence $\frac{u_x}{c_x} = \sum_y \frac{u_y}{c_y} p_{xy}$. It follows that $\frac{u_x}{c_x}$ is harmonic with a and b as the boundary where the boundary conditions are $u_b = 0$ and u_a equals some fixed value. Now, $\frac{u_b}{c_b} = 0$. Setting the current into a to one, fixed the value of v_a. Adjust the current into a so that v_a equals $\frac{u_a}{c_a}$. Now $\frac{u_x}{c_x}$ and v_x satisfy the same harmonic conditions and thus are the same harmonic function. Let the current into a correspond to one walk. Note that if the walk starts at a and ends at b, the expected value of the difference between the number of times the walk leaves a and enters a must be one. This implies that the amount of current into a corresponds to one walk.

Next we need to show that the current i_{xy} is the net frequency with which a random walk traverses edge xy.

$$i_{xy} = (v_x - v_y) c_{xy} = \left(\frac{u_x}{c_x} - \frac{u_y}{c_y} \right) c_{xy} = u_x \frac{c_{xy}}{c_x} - u_y \frac{c_{xy}}{c_y} = u_x p_{xy} - u_y p_{yx}$$

The quantity $u_x p_{xy}$ is the expected number of times the edge xy is traversed from x to y and the quantity $u_y p_{yx}$ is the expected number of times the edge xy is traversed from y to

x. Thus, the current i_{xy} is the expected net number of traversals of the edge xy from x to y.

Effective resistance and escape probability

Set $v_a = 1$ and $v_b = 0$. Let i_a be the current flowing into the network at vertex a and out at vertex b. Define the *effective resistance* r_{eff} between a and b to be $r_{eff} = \frac{v_a}{i_a}$ and the *effective conductance* c_{eff} to be $c_{eff} = \frac{1}{r_{eff}}$. Define the *escape probability*, p_{escape}, to be the probability that a random walk starting at a reaches b before returning to a. We now show that the escape probability is $\frac{c_{eff}}{c_a}$. For convenience, assume that a and b are not adjacent. A slight modification of our argument suffices for the case when a and b are adjacent.

$$i_a = \sum_y (v_a - v_y)c_{ay}$$

Since $v_a = 1$,

$$\begin{aligned} i_a &= \sum_y c_{ay} - c_a \sum_y v_y \frac{c_{ay}}{c_a} \\ &= c_a \left[1 - \sum_y p_{ay} v_y\right]. \end{aligned}$$

For each y adjacent to the vertex a, p_{ay} is the probability of the walk going from vertex a to vertex y. Earlier we showed that v_y is the probability of a walk starting at y going to a before reaching b. Thus, $\sum_y p_{ay} v_y$ is the probability of a walk starting at a returning to a before reaching b and $1 - \sum_y p_{ay} v_y$ is the probability of a walk starting at a reaching b before returning to a. Thus, $i_a = c_a p_{escape}$. Since $v_a = 1$ and $c_{eff} = \frac{i_a}{v_a}$, it follows that $c_{eff} = i_a$. Thus, $c_{eff} = c_a p_{escape}$ and hence $p_{escape} = \frac{c_{eff}}{c_a}$.

For a finite connected graph the escape probability will always be nonzero. Now consider an infinite graph such as a lattice and a random walk starting at some vertex a. Form a series of finite graphs by merging all vertices at distance d or greater from a into a single vertex b for larger and larger values of d. The limit of p_{escape} as d goes to infinity is the probability that the random walk will never return to a. If $p_{escape} \to 0$, then eventually any random walk will return to a. If $p_{escape} \to q$ where $q > 0$, then a fraction of the walks never return. Thus, the escape probability terminology.

5.3 Random Walks on Undirected Graphs with Unit Edge Weights

We now focus our discussion on random walks on undirected graphs with uniform edge weights. At each vertex, the random walk is equally likely to take any edge. This corresponds to an electrical network in which all edge resistances are one. Assume the graph is connected. We consider questions such as what is the expected time for a random

walk starting at a vertex x to reach a target vertex y, what is the expected time until the random walk returns to the vertex it started at, and what is the expected time to reach every vertex?

Hitting time

The *hitting time* h_{xy}, sometimes called *discovery time*, is the expected time of a random walk starting at vertex x to reach vertex y. Sometimes a more general definition is given where the hitting time is the expected time to reach a vertex y from a given starting probability distribution.

One interesting fact is that adding edges to a graph may either increase or decrease h_{xy} depending on the particular situation. Adding an edge can shorten the distance from x to y thereby decreasing h_{xy} or the edge could increase the probability of a random walk going to some far off portion of the graph thereby increasing h_{xy}. Another interesting fact is that hitting time is not symmetric. The expected time to reach a vertex y from a vertex x in an undirected graph may be radically different from the time to reach x from y.

We start with two technical lemmas. The first lemma states that the expected time to traverse a path of n vertices is $\Theta(n^2)$.

Lemma 5.4 *The expected time for a random walk starting at one end of a path of n vertices to reach the other end is $\Theta(n^2)$.*

Proof: Consider walking from vertex 1 to vertex n in a graph consisting of a single path of n vertices. Let h_{ij}, $i < j$, be the hitting time of reaching j starting from i. Now $h_{12} = 1$ and

$$h_{i,i+1} = \tfrac{1}{2} + \tfrac{1}{2}(1 + h_{i-1,i+1}) = 1 + \tfrac{1}{2}(h_{i-1,i} + h_{i,i+1}) \quad 2 \le i \le n-1.$$

Solving for $h_{i,i+1}$ yields the recurrence

$$h_{i,i+1} = 2 + h_{i-1,i}.$$

Solving the recurrence yields

$$h_{i,i+1} = 2i - 1.$$

To get from 1 to n, go from 1 to 2, 2 to 3, etc. Thus

$$\begin{aligned} h_{1,n} &= \sum_{i=1}^{n-1} h_{i,i+1} = \sum_{i=1}^{n-1} (2i-1) \\ &= 2\sum_{i=1}^{n-1} i - \sum_{i=1}^{n-1} 1 \\ &= 2\frac{n(n-1)}{2} - (n-1) \\ &= (n-1)^2. \end{aligned}$$

∎

The lemma says that in a random walk on a line where we are equally likely to take one step to the right or left each time, the farthest we will go away from the start in n steps is $\Theta(\sqrt{n})$.

The next lemma shows that the expected time spent at vertex i by a random walk from vertex 1 to vertex n in a chain of n vertices is $2(i-1)$ for $2 \leq i \leq n-1$.

Lemma 5.5 *Consider a random walk from vertex 1 to vertex n in a chain of n vertices. Let t(i) be the expected time spent at vertex i. Then*

$$t(i) = \begin{cases} n-1 & i = 1 \\ 2(n-i) & 2 \leq i \leq n-1 \\ 1 & i = n. \end{cases}$$

Proof: Now $t(n) = 1$ since the walk stops when it reaches vertex n. Half of the time when the walk is at vertex $n-1$ it goes to vertex n. Thus $t(n-1) = 2$. For $3 \leq i < n-1$, $t(i) = \frac{1}{2}[t(i-1) + t(i+1)]$ and $t(1)$ and $t(2)$ satisfy $t(1) = \frac{1}{2}t(2) + 1$ and $t(2) = t(1) + \frac{1}{2}t(3)$. Solving for $t(i+1)$ for $3 \leq i < n-1$ yields

$$t(i+1) = 2t(i) - t(i-1)$$

which has solution $t(i) = 2(n-i)$ for $3 \leq i < n-1$. Then solving for $t(2)$ and $t(1)$ yields $t(2) = 2(n-2)$ and $t(1) = n-1$. Thus, the total time spent at vertices is

$$n - 1 + 2(1 + 2 + \cdots + n - 2) + 1 = (n-1) + 2\frac{(n-1)(n-2)}{2} + 1 = (n-1)^2 + 1$$

which is one more than h_{1n} and thus is correct. ∎

Adding edges to a graph might either increase or decrease the hitting time h_{xy}. Consider the graph consisting of a single path of n vertices. Add edges to this graph to get the graph in Figure 5.3 consisting of a clique of size $n/2$ connected to a path of $n/2$ vertices. Then add still more edges to get a clique of size n. Let x be the vertex at the midpoint of the original path and let y be the other endpoint of the path consisting of $n/2$ vertices as shown in the figure. In the first graph consisting of a single path of length n, $h_{xy} = \Theta(n^2)$. In the second graph consisting of a clique of size $n/2$ along with a path of length $n/2$, $h_{xy} = \Theta(n^3)$. To see this latter statement, note that starting at x, the walk will go down the path towards y and return to x $n/2$ times on average before reaching y for the first time. Each time the walk in the path returns to x, with probability $(n/2-1)/(n/2)$ it enters the clique and thus on average enters the clique $\Theta(n)$ times before starting down the path again. Each time it enters the clique, it spends $\Theta(n)$ time in the clique before returning to x. Thus, each time the walk returns to x from the path it spends $\Theta(n^2)$ time in the clique before starting down the path towards y for a total expected time that is

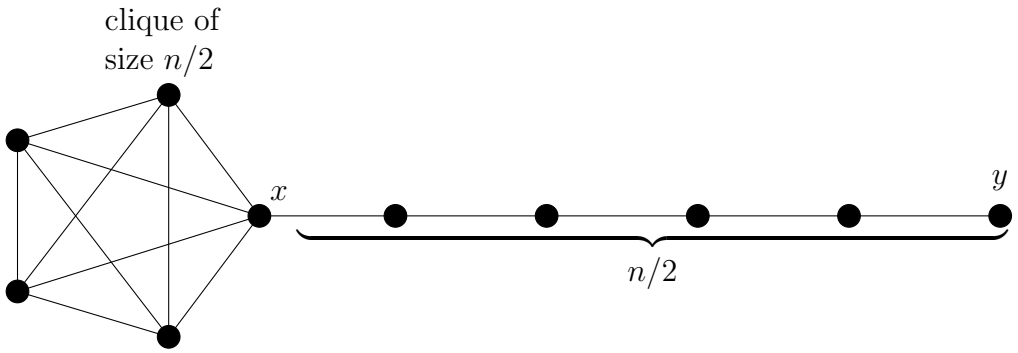


Figure 5.3: Illustration that adding edges to a graph can either increase or decrease hitting time.

$\Theta(n^3)$ before reaching y. In the third graph, which is the clique of size n, $h_{xy} = \Theta(n)$. Thus, adding edges first increased h_{xy} from n^2 to n^3 and then decreased it to n.

Hitting time is not symmetric even in the case of undirected graphs. In the graph of Figure 5.3, the expected time, h_{xy}, of a random walk from x to y, where x is the vertex of attachment and y is the other end vertex of the chain, is $\Theta(n^3)$. However, h_{yx} is $\Theta(n^2)$.

Commute time

The *commute time*, commute(x, y), is the expected time of a random walk starting at x reaching y and then returning to x. So commute$(x, y) = h_{xy} + h_{yx}$. Think of going from home to office and returning home. We now relate the commute time to an electrical quantity, the effective resistance. The *effective resistance* between two vertices x and y in an electrical network is the voltage difference between x and y when one unit of current is inserted at vertex x and withdrawn from vertex y.

Theorem 5.6 *Given an undirected graph, consider the electrical network where each edge of the graph is replaced by a one ohm resistor. Given vertices x and y, the commute time, commute(x, y), equals $2mr_{xy}$ where r_{xy} is the effective resistance from x to y and m is the number of edges in the graph.*

Proof: Insert at each vertex i a current equal to the degree d_i of vertex i. The total current inserted is $2m$ where m is the number of edges. Extract from a specific vertex j all of this $2m$ current. Let v_{ij} be the voltage difference from i to j. The current into i divides into the d_i resistors at vertex i. The current in each resistor is proportional to the voltage across it. Let k be a vertex adjacent to i. Then the current through the resistor between i and k is $v_{ij} - v_{kj}$, the voltage drop across the resister. The sum of the currents out of i through the resisters must equal d_i, the current injected into i.

$$d_i = \sum_{\substack{k \text{ adj} \\ \text{to } i}} (v_{ij} - v_{kj}) = d_i v_{ij} - \sum_{\substack{k \text{ adj} \\ \text{to } i}} v_{kj}.$$

Solving for v_{ij}

$$v_{ij} = 1 + \sum_{\substack{k \text{ adj} \\ \text{to } i}} \frac{1}{d_i} v_{kj} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \frac{1}{d_i}(1 + v_{kj}). \tag{5.1}$$

Now the hitting time from i to j is the average time over all paths from i to k adjacent to i and then on from k to j. This is given by

$$h_{ij} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \frac{1}{d_i}(1 + h_{kj}). \tag{5.2}$$

Subtracting (5.2) from (5.1), gives $v_{ij} - h_{ij} = \sum_{\substack{k \text{ adj} \\ \text{to } i}} \frac{1}{d_i}(v_{kj} - h_{kj})$. Thus, the function $v_{ij} - h_{ij}$ is harmonic. Designate vertex j as the only boundary vertex. The value of $v_{ij} - h_{ij}$ at $i = j$, namely $v_{jj} - h_{jj}$, is zero, since both v_{jj} and h_{jj} are zero. So the function $v_{ij} - h_{ij}$ must be zero everywhere. Thus, the voltage v_{ij} equals the expected time h_{ij} from i to j.

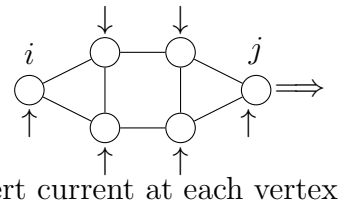
To complete the proof, note that $h_{ij} = v_{ij}$ is the voltage from i to j when currents are inserted at all vertices in the graph and extracted at vertex j. If the current is extracted from i instead of j, then the voltages change and $v_{ji} = h_{ji}$ in the new setup. Finally, reverse all currents in this latter step. The voltages change again and for the new voltages $-v_{ji} = h_{ji}$. Since $-v_{ji} = v_{ij}$, we get $h_{ji} = v_{ij}$.

Thus, when a current is inserted at each vertex equal to the degree of the vertex and the current is extracted from j, the voltage v_{ij} in this set up equals h_{ij}. When we extract the current from i instead of j and then reverse all currents, the voltage v_{ij} in this new set up equals h_{ji}. Now, superpose both situations, i.e., add all the currents and voltages. By linearity, for the resulting v_{ij}, which is the sum of the other two v_{ij}'s, is $v_{ij} = h_{ij} + h_{ji}$. All currents cancel except the $2m$ amps injected at i and withdrawn at j. Thus, $2mr_{ij} = v_{ij} = h_{ij} + h_{ji} = \text{commute}(i, j)$ or $\text{commute}(i, j) = 2mr_{ij}$ where r_{ij} is the effective resistance from i to j. ∎

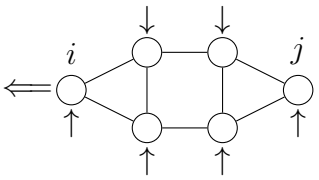
The following corollary follows from Theorem 5.6 since the effective resistance r_{uv} is less than or equal to one when u and v are connected by an edge.

Corollary 5.7 *If vertices x and y are connected by an edge, then $h_{xy} + h_{yx} \leq 2m$ where m is the number of edges in the graph.*

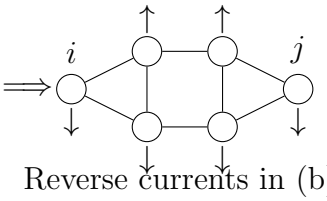
Proof: If x and y are connected by an edge, then the effective resistance r_{xy} is less than or equal to one. ∎



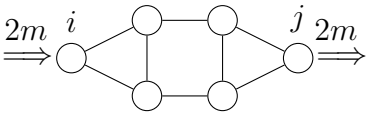
Insert current at each vertex
equal to degree of vertex.
Extract $2m$ at vertex j.
$v_{ij} = h_{ij}$
(a)



Extract current from i instead of j.
For new voltages $v_{ji} = h_{ji}$.
(b)



Reverse currents in (b).
For new voltages $-v_{ji} = h_{ji.}$
Since $-v_{ji} = v_{ij}, h_{ji} = v_{ij}$.
(c)



Superpose currents in (a) and (c).
$2mr_{ij} = v_{ij} = h_{ij} + h_{ji} = commute(i, j)$
(d)

Figure 5.4: Illustration of proof that commute$(x, y) = 2mr_{xy}$ where m is the number of edges in the undirected graph and r_{xy} is the effective resistance between x and y.

Corollary 5.8 *For vertices x and y in an n vertex graph, the commute time, commute(x, y), is less than or equal to n^3.*

Proof: By Theorem 5.6 the commute time is given by the formula commute$(x, y) = 2mr_{xy}$ where m is the number of edges. In an n vertex graph there exists a path from x to y of length at most n. This implies $r_{xy} \leq n$ since the resistance can not be greater than that of any path from x to y. Since the number of edges is at most $\binom{n}{2}$

$$\text{commute}(x, y) = 2mr_{xy} \leq 2\binom{n}{2} n \cong n^3.$$

■

Again adding edges to a graph may increase or decrease the commute time. To see this, consider the graph consisting of a chain of n vertices, the graph of Figure 5.3, and the clique on n vertices.

Cover time

The *cover time*, cover(x, G) , is the expected time of a random walk starting at vertex x in the graph G to reach each vertex at least once. We write cover(x) when G is understood.

The cover time of an undirected graph G, denoted cover(G), is

$$\text{cover}(G) = \max_x \text{cover}(x, G).$$

For cover time of an undirected graph, increasing the number of edges in the graph may increase or decrease the cover time depending on the situation. Again consider three graphs, a chain of length n which has cover time $\Theta(n^2)$, the graph in Figure 5.3 which has cover time $\Theta(n^3)$, and the complete graph on n vertices which has cover time $\Theta(n \log n)$. Adding edges to the chain of length n to create the graph in Figure 5.3 increases the cover time from n^2 to n^3 and then adding even more edges to obtain the complete graph reduces the cover time to $n \log n$.

Note: The cover time of a clique is $\theta(n \log n)$ since this is the time to select every integer out of n integers with high probability, drawing integers at random. This is called the *coupon collector problem*. The cover time for a straight line is $\Theta(n^2)$ since it is the same as the hitting time. For the graph in Figure 5.3, the cover time is $\Theta(n^3)$ since one takes the maximum over all start states and $\text{cover}(x, G) = \Theta(n^3)$ where x is the vertex of attachment.

Theorem 5.9 *Let G be a connected graph with n vertices and m edges. The time for a random walk to cover all vertices of the graph G is bounded above by $4m(n-1)$.*

Proof: Consider a depth first search of the graph G starting from some vertex z and let T be the resulting depth first search spanning tree of G. The depth first search covers every vertex. Consider the expected time to cover every vertex in the order visited by the depth first search. Clearly this bounds the cover time of G starting from vertex z. Note that each edge in T is traversed twice, once in each direction.

$$\text{cover}(z, G) \leq \sum_{\substack{(x,y)\in T \\ (y,x)\in T}} h_{xy}.$$

If (x, y) is an edge in T, then x and y are adjacent and thus Corollary 5.7 implies $h_{xy} \leq 2m$. Since there are $n-1$ edges in the dfs tree and each edge is traversed twice, once in each direction, $\text{cover}(z) \leq 4m(n-1)$. This holds for all starting vertices z. Thus, $\text{cover}(G) \leq 4m(n-1)$ ∎

The theorem gives the correct answer of n^3 for the $n/2$ clique with the $n/2$ tail. It gives an upper bound of n^3 for the n-clique where the actual cover time is $n \log n$.

Let r_{xy} be the effective resistance from x to y. Define the resistance $r_{eff}(G)$ of a graph G by $r_{eff}(G) = \max_{x,y}(r_{xy})$.

Theorem 5.10 *Let G be an undirected graph with m edges. Then the cover time for G is bounded by the following inequality*

$$mr_{eff}(G) \leq cover(G) \leq 2e^3 mr_{eff}(G) \ln n + n$$

where e=2.71 is Euler's constant and $r_{eff}(G)$ is the resistance of G.

Proof: By definition $r_{eff}(G) = \max_{x,y}(r_{xy})$. Let u and v be the vertices of G for which r_{xy} is maximum. Then $r_{eff}(G) = r_{uv}$. By Theorem 5.6, commute$(u, v) = 2mr_{uv}$. Hence $mr_{uv} = \frac{1}{2}$commute(u, v). Clearly the commute time from u to v and back to u is less than twice the $\max(h_{uv}, h_{vu})$ and $\max(h_{uv}, h_{vu})$ is clearly less than the cover time of G. Putting these facts together gives the first inequality in the theorem.

$$mr_{eff}(G) = mr_{uv} = \tfrac{1}{2}\text{commute}(u, v) \leq \max(h_{uv}, h_{vu}) \leq \text{cover}(G)$$

For the second inequality in the theorem, by Theorem 5.6, for any x and y, commute(x, y) equals $2mr_{xy}$ which is less than or equal to $2mr_{eff}(G)$, implying $h_{xy} \leq 2mr_{eff}(G)$. By the Markov inequality, since the expected time to reach y starting at any x is less than $2mr_{eff}(G)$, the probability that y is not reached from x in $2mr_{eff}(G)e^3$ steps is at most $\frac{1}{e^3}$. Thus, the probability that a vertex y has not been reached in $2e^3 mr_{eff}(G) \log n$ steps is at most $\frac{1}{e^3}^{\ln n} = \frac{1}{n^3}$ because a random walk of length $2e^3 mr(G) \log n$ is a sequence of $\log n$ independent random walks, each of length $2e^3 mr(G)r_{eff}(G)$. Suppose after a walk of $2e^3 mr_{eff}(G) \log n$ steps, vertices $v_1, v_2, \ldots, v_l$ had not been reached. Walk until v_1 is reached, then v_2, etc. By Corollary 5.8 the expected time for each of these is n^3, but since each happens only with probability $1/n^3$, we effectively take $O(1)$ time per v_i, for a total time at most n. More precisely,

$$\begin{aligned}\text{cover}(G) &\leq 2e^3 mr_{eff}(G) \log n + \sum_v \text{Prob}\left(v \text{ was not visited in the first } 2e^3 mr_{eff}(G) \text{ steps}\right) n^3 \\ &\leq 2e^3 mr_{eff}(G) \log n + \sum_v \frac{1}{n^3} n^3 \leq 2e^3 mr_{eff}(G) + n.\end{aligned}$$

■

5.4 Random Walks in Euclidean Space

Many physical processes such as Brownian motion are modeled by random walks. Random walks in Euclidean d-space consisting of fixed length steps parallel to the coordinate axes are really random walks on a d-dimensional lattice and are a special case of random walks on graphs. In a random walk on a graph, at each time unit an edge from the current vertex is selected at random and the walk proceeds to the adjacent vertex. We begin by studying random walks on lattices.

Random walks on lattices

We now apply the analogy between random walks and current to lattices. Consider a random walk on a finite segment $-n, \ldots, -1, 0, 1, 2, \ldots, n$ of a one dimensional lattice starting from the origin. Is the walk certain to return to the origin or is there some probability that it will escape, i.e., reach the boundary before returning? The probability of reaching the boundary before returning to the origin is called the escape probability. We shall be interested in this quantity as n goes to infinity.

Convert the lattice to an electrical network by replacing each edge with a one ohm resister. Then the probability of a walk starting at the origin reaching n or $-n$ before returning to the origin is the escape probability given by

$$p_{escape} = \frac{c_{eff}}{c_a}$$

where c_{eff} is the effective conductance between the origin and the boundary points and c_a is the sum of the conductance's at the origin. In a d-dimensional lattice, $c_a = 2d$ assuming that the resistors have value one. For the d-dimensional lattice

$$p_{escape} = \frac{1}{2d\, r_{eff}}$$

In one dimension, the electrical network is just two series connections of n one ohm resistors connected in parallel. So as n goes to infinity, r_{eff} goes to infinity and the escape probability goes to zero as n goes to infinity. Thus, the walk in the unbounded one dimensional lattice will return to the origin with probability one. This is equivalent to flipping a balanced coin and keeping tract of the number of heads minus the number of tails. The count will return to zero infinitely often. By the law of large numbers in n steps with high probability the walk will be within $\sqrt{n}$ distance of the origin.

Two dimensions

For the 2-dimensional lattice, consider a larger and larger square about the origin for the boundary as shown in Figure 5.5a and consider the limit of r_{eff} as the squares get larger. Shorting the resistors on each square can only reduce r_{eff}. Shorting the resistors results in the linear network shown in Figure 5.5b. As the paths get longer, the number of resistors in parallel also increases. The resistor between vertex i and $i+1$ is really $4(2i+1)$ unit resistors in parallel. The effective resistance of $4(2i+1)$ resistors in parallel is $1/4(2i+1)$. Thus,

$$r_{eff} \geq \tfrac{1}{4} + \tfrac{1}{12} + \tfrac{1}{20} + \cdots = \tfrac{1}{4}(1 + \tfrac{1}{3} + \tfrac{1}{5} + \cdots) = \Theta(\ln n).$$

Since the lower bound on the effective resistance and hence the effective resistance goes to infinity, the escape probability goes to zero for the 2-dimensional lattice.

Three dimensions

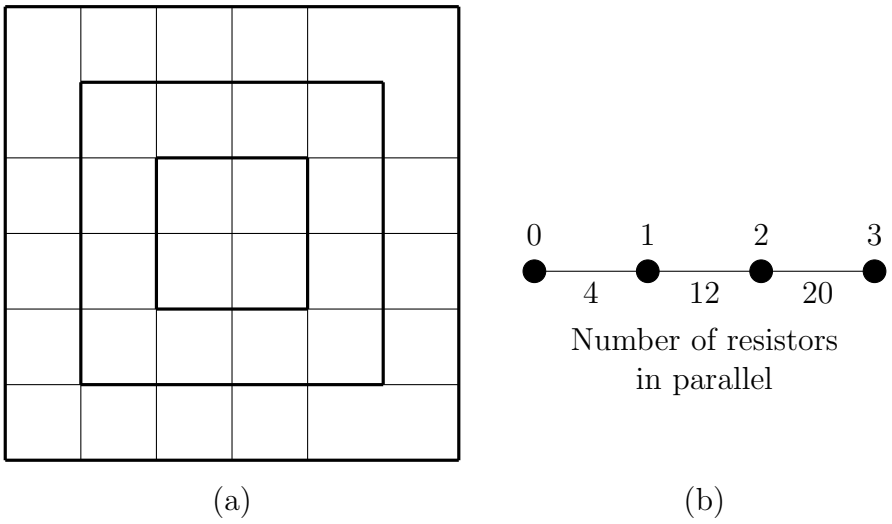


Figure 5.5: 2-dimensional lattice along with the linear network resulting from shorting resistors on the concentric squares about the origin.

In three dimensions, the resistance along any path to infinity grows to infinity but the number of paths in parallel also grows to infinity. It turns out there are a sufficient number of paths that r_{eff} remains finite and thus there is a nonzero escape probability. We will prove this now. First note that shorting any edge decreases the resistance, so we do not use shorting in this proof, since we seek to prove an upper bound on the resistance. Instead we remove some edges, which increases their resistance to infinity and hence increases the effective resistance, giving an upper bound. To simplify things we consider walks on on quadrant rather than the full grid. The resistance to infinity derived from only the quadrant is an upper bound on the resistance of the full grid.

The construction used in three dimensions is easier to explain first in two dimensions. Draw dotted diagonal lines at $x + y = 2^n - 1$. Consider two paths that start at the origin. One goes up and the other goes to the right. Each time a path encounters a dotted diagonal line, split the path into two, one which goes right and the other up. Where two paths cross, split the vertex into two, keeping the paths separate. By a symmetry argument, splitting the vertex does not change the resistance of the network. Remove all resistors except those on these paths. The resistance of the original network is less than that of the tree produced by this process since removing a resistor is equivalent to increasing its resistance to infinity.

The distances between splits increase and are 1, 2, 4, etc. At each split the number of paths in parallel doubles. See Figure 5.7. Thus, the resistance to infinity in this two dimensional example is

$$\frac{1}{2} + \frac{1}{4}2 + \frac{1}{8}4 + \cdots = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots = \infty.$$

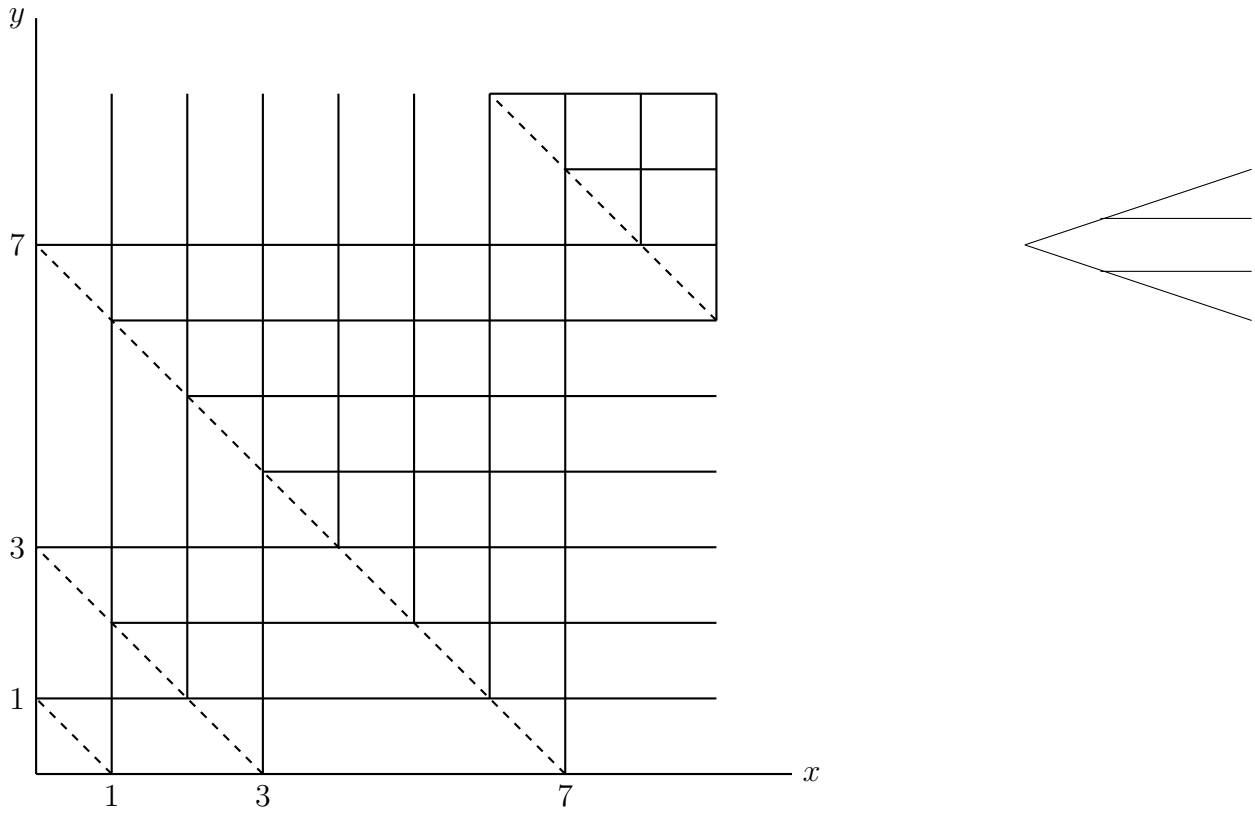


Figure 5.6: Paths in a 2-dimensional lattice obtained from the 3-dimensional construction applied in 2-dimensions.

In the analogous three dimensional construction, paths go up, to the right, and out of the plane of the paper. The paths split three ways at planes given by $x + y + z = 2^n - 1$. Each time the paths split the number of parallel segments triple. Segments of the paths between splits are of length 1, 2, 4, etc. and the resistance of the segments are equal to the lengths. The resistance out to infinity for the tree is

$$\tfrac{1}{3} + \tfrac{1}{9}2 + \tfrac{1}{27}4 + \cdots = \tfrac{1}{3}\left(1 + \tfrac{2}{3} + \tfrac{4}{9} + \cdots\right) = \tfrac{1}{3}\frac{1}{1-\frac{2}{3}} = 1$$

The resistance of the three dimensional lattice is less. It is important to check that the paths are edge-disjoint and so the tree is a subgraph of the lattice. Going to a subgraph is equivalent to deleting edges which only increases the resistance. That is why the resistance of the lattice is less than that of the tree. Thus, in three dimensions the escape probability is nonzero. The upper bound on r_{eff} gives the lower bound

$$p_{escape} = \tfrac{1}{2d}\tfrac{1}{r_{eff}} \geq \tfrac{1}{6}.$$

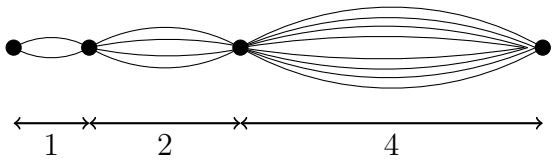


Figure 5.7: Paths obtained from 2-dimensional lattice. Distances between splits double as do the number of parallel paths.

A lower bound on r_{eff} gives an upper bound on p_{escape}. To get the upper bound on p_{escape}, short all resistors on surfaces of boxes at distances $1, 2, 3,$, etc. Then

$$r_{eff} \geq \frac{1}{6}\left[1 + \frac{1}{9} + \frac{1}{25} + \cdots\right] \geq \frac{1.23}{6} \geq 0.2$$

This gives

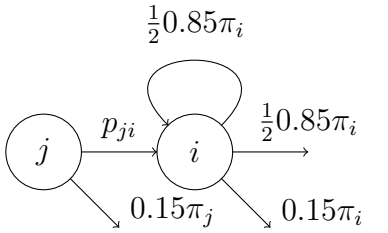
$$p_{escape} = \frac{1}{2d}\frac{1}{r_{eff}} \leq \frac{5}{6}.$$

5.5 The Web as a Markov Chain

A modern application of random walks on directed graphs comes from trying to establish the importance of pages on the World Wide Web. One way to do this would be to take a random walk on the web viewed as a directed graph with an edge corresponding to each hypertext link and rank pages according to their stationary probability. A connected, undirected graph is strongly connected in that one can get from any vertex to any other vertex and back again. Often the directed case is not strongly connected. One difficulty occurs if there is a vertex with no out edges. When the walk encounters this vertex the walk disappears. Another difficulty is that a vertex or a strongly connected component with no in edges is never reached. One way to resolve these difficulties is to introduce a random restart condition. At each step, with some probability r, jump to a vertex selected uniformly at random and with probability $1 - r$ select an edge at random and follow it. If a vertex has no out edges, the value of r for that vertex is set to one. This has the effect of converting the graph to a strongly connected graph so that the stationary probabilities exist.

Page rank and hitting time

The page rank of a vertex in a directed graph is the stationary probability of the vertex, where we assume a positive restart probability of say $r = 0.15$. The restart ensures that the graph is strongly connected. The page rank of a page is the fractional frequency with which the page will be visited over a long period of time. If the page rank is p, then the expected time between visits or return time is $1/p$. Notice that one can increase the pagerank of a page by reducing the return time and this can be done by creating short cycles.



$$\pi_i = 0.85\pi_j p_{ji} + \tfrac{0.85}{2}\pi_i$$

$$\pi_i = 1.48\pi_j p_{ji}$$

Figure 5.8: Impact on page rank of adding a self loop

Consider a vertex i with a single edge in from vertex j and a single edge out. The stationary probability $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi}P = \boldsymbol{\pi}$, and thus

$$\pi_i = \pi_j p_{ji}.$$

Adding a self-loop at i, results in a new equation

$$\pi_i = \pi_j p_{ji} + \frac{1}{2}\pi_i$$

or

$$\pi_i = 2\ \pi_j p_{ji}.$$

Of course, π_j would have changed too, but ignoring this for now, pagerank is doubled by the addition of a self-loop. Adding k self loops, results in the equation

$$\pi_i = \pi_j p_{ji} + \frac{k}{k+1}\pi_i,$$

and again ignoring the change in π_j, we now have $\pi_i = (k+1)\pi_j p_{ji}$. What prevents one from increasing the page rank of a page arbitrarily? The answer is the restart. We neglected the 0.15 probability that is taken off for the random restart. With the restart taken into account, the equation for π_i when there is no self-loop is

$$\pi_i = 0.85\pi_j p_{ji}$$

whereas, with k self-loops, the equation is

$$\pi_i = 0.85\pi_j p_{ji} + 0.85\frac{k}{k+1}\pi_i.$$

Solving for π_i yields

$$\pi_i = \frac{0.85k + 0.85}{0.15k + 1}\pi_j p_{ji}$$

which for $k = 1$ is $\pi_i = 1.48\pi_j P_{ji}$ and in the limit as $k \to \infty$ is $\pi_i = 5.67\pi_j p_{ji}$. Adding a single loop only increases pagerank by a factor of 1.74 and adding k loops increases it by at most a factor of 6.67 for arbitrarily large k.

Hitting time

Related to page rank is a quantity called hitting time. Hitting time is closely related to return time and thus to the reciprocal of page rank. One way to return to a vertex v is by a path in the graph from v back to v. Another way is to start on a path that encounters a restart, followed by a path from the random restart vertex to v. The time to reach v after a restart is the hitting time. Thus, return time is clearly less than the expected time until a restart plus hitting time. The fastest one could return would be if there were only paths of length two since self loops are ignored in calculating page rank. If r is the restart value, then the loop would be traversed with at most probability $(1-r)^2$. With probability $r + (1-r)\, r = (2-r)\, r$ one restarts and then hits v. Thus, the return time is at least $2\,(1-r)^2 + (2-r)\, r \times$ (hitting time). Combining these two bounds yields

$$2\,(1-r)^2 + (2-r)\, rE\,(\text{hitting time}) \leq E\,(\text{return time}) \leq E\,(\text{hitting time})\,.$$

The relationship between return time and hitting time can be used to see if a vertex has unusually high probability of short loops. However, there is no efficient way to compute hitting time for all vertices as there is for return time. For a single vertex v, one can compute hitting time by removing the edges out of the vertex v for which one is computing hitting time and then run the page rank algorithm for the new graph. The hitting time for v is the reciprocal of the page rank in the graph with the edges out of v removed. Since computing hitting time for each vertex requires removal of a different set of edges, the algorithm only gives the hitting time for one vertex at a time. Since one is probably only interested in the hitting time of vertices with low hitting time, an alternative would be to use a random walk to estimate the hitting time of low hitting time vertices.

Spam

Suppose one has a web page and would like to increase its page rank by creating some other web pages with pointers to the original page. The abstract problem is the following. We are given a directed graph G and a vertex v whose page rank we want to increase. We may add new vertices to the graph and add edges from v or from the new vertices to any vertices we want. We cannot add edges out of other vertices. We can also delete edges from v.

The page rank of v is the stationary probability for vertex v with random restarts. If we delete all existing edges out of v, create a new vertex u and edges (v, u) and (u, v), then the page rank will be increased since any time the random walk reaches v it will be captured in the loop $v \to u \to v$. A search engine can counter this strategy by more frequent random restarts.

A second method to increase page rank would be to create a star consisting of the vertex v at its center along with a large set of new vertices each with a directed edge to

v. These new vertices will sometimes be chosen as the target of the random restart and hence the vertices increase the probability of the random walk reaching v. This second method is countered by reducing the frequency of random restarts.

Notice that the first technique of capturing the random walk increases page rank but does not effect hitting time. One can negate the impact of someone capturing the random walk on page rank by increasing the frequency of random restarts. The second technique of creating a star increases page rank due to random restarts and decreases hitting time. One can check if the page rank is high and hitting time is low in which case the page rank is likely to have been artificially inflated by the page capturing the walk with short cycles.

Personalized page rank

In computing page rank, one uses a restart probability, typically 0.15, in which at each step, instead of taking a step in the graph, the walk goes to a vertex selected uniformly at random. In personalized page rank, instead of selecting a vertex uniformly at random, one selects a vertex according to a personalized probability distribution. Often the distribution has probability one for a single vertex and whenever the walk restarts it restarts at that vertex.

Algorithm for computing personalized page rank

First, consider the normal page rank. Let α be the restart probability with which the random walk jumps to an arbitrary vertex. With probability $1-\alpha$ the random walk selects a vertex uniformly at random from the set of adjacent vertices. Let $\mathbf{p}$ be a row vector denoting the page rank and let G be the adjacency matrix with rows normalized to sum to one. Then

$$\mathbf{p} = \tfrac{\alpha}{n}(1,1,\ldots,1) + (1-\alpha)\,\mathbf{p}G$$

$$\mathbf{p}[I-(1-\alpha)G] = \frac{\alpha}{n}(1,1,\ldots,1)$$

or

$$\mathbf{p} = \tfrac{\alpha}{n}(1,1,\ldots,1)\,[I-(1-\alpha)\,G]^{-1}.$$

Thus, in principle, $\mathbf{p}$ can be found by computing the inverse of $[I-(1-\alpha)G]^{-1}$. But this is far from practical since for the whole web one would be dealing with matrices with billions of rows and columns. A more practical procedure is to run the random walk and observe using the basics of the power method in Chapter 3 that the process converges to the solution $\mathbf{p}$.

For the personalized page rank, instead of restarting at an arbitrary vertex, the walk restarts at a designated vertex. More generally, it may restart in some specified neighborhood. Suppose the restart selects a vertex using the probability distribution s. Then, in

the above calculation replace the vector $\frac{1}{n}(1,1,\ldots,1)$ by the vector $\mathbf{s}$. Again, the computation could be done by a random walk. But, we wish to do the random walk calculation for personalized pagerank quickly since it is to be performed repeatedly. With more care this can be done, though we do not describe it here.

5.6 Markov Chain Monte Carlo

The Markov Chain Monte Carlo (MCMC) method is a technique for sampling a multivariate probability distribution $p(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$. The MCMC method is used to estimate the expected value of a function $f(\mathbf{x})$

$$E(f) = \sum_{\mathbf{x}} f(\mathbf{x})p(\mathbf{x}).$$

If each x_i can take on two or more values, then there are at least 2^d values for $\mathbf{x}$, so an explicit summation requires exponential time. Instead, one could draw a set of samples, each sample $\mathbf{x}$ with probability $p(\mathbf{x})$. Averaging f over these samples provides an estimate of the sum.

To sample according to $p(\mathbf{x})$, design a Markov Chain whose states correspond to the possible values of $\mathbf{x}$ and whose stationary probability distribution is $p(\mathbf{x})$. There are two general techniques to design such a Markov Chain: the Metropolis-Hastings algorithm and Gibbs sampling. The Fundamental Theorem of Markov Chains, Theorem 5.2, states that the average of f over states seen in a sufficiently long run is a good estimate of $E(f)$. The harder task is to show that the number of steps needed before the long-run average probabilities are close to the stationary distribution grows polynomially in d, though the total number of states may grow exponentially in d. This phenomenon known as *rapid mixing* happens for a number of interesting examples. Section 5.8 presents a crucial tool used to show rapid mixing.

We used $\mathbf{x} \in \mathbf{R}^d$ to emphasize that distributions are multi-variate. From a Markov chain perspective, each value $\mathbf{x}$ can take on is a state, i.e., a vertex of the graph on which the random walk takes place. Henceforth, we will use the subscripts $i, j, k, \ldots$ to denote states and will use p_i instead of $p(x_1, x_2, \ldots, x_d)$ to denote the probability of the state corresponding to a given set of values for the variables. Recall that in the Markov chain terminology, vertices of the graph are called states.

Recall the notation that $\mathbf{p^{(t)}}$ is the row vector of probabilities of the random walk being at each state (vertex of the graph) at time t. So, $\mathbf{p^{(t)}}$ has as many components as there are states and its i^{th} component, $p_i^{(t)}$, is the probability of being in state i at time t. Recall the long-term t-step average is

$$\mathbf{a^{(t)}} = \frac{1}{t}\left[\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}}\right]. \tag{5.3}$$

The expected value of the function f under the probability distribution $\mathbf{p}$ is $E(f) = \sum_i f_i p_i$ where f_i is the value of f at state i. Our estimate of this quantity will be the average value of f at the states seen in a t step walk. Call this estimate a. Clearly, the expected value of a is

$$E(a) = \sum_i f_i \left(\frac{1}{t} \sum_{j=1}^{t} \text{Prob (walk is in state } i \text{ at time } j) \right) = \sum_i f_i a_i^{(t)}.$$

The expectation here is with respect to the "coin tosses" of the algorithm, not with respect to the underlying distribution p. Let $f_{\max}$ denote the maximum absolute value of f. It is easy to see that

$$\left| \sum_i f_i p_i - E(a) \right| \leq f_{\max} \sum_i |p_i - a_i^{(t)}| = f_{\max} |\mathbf{p} - \mathbf{a^{(t)}}|_1 \tag{5.4}$$

where the quantity $|\mathbf{p} - \mathbf{a^{(t)}}|_1$ is the l_1 distance between the probability distributions $\mathbf{p}$ and $\mathbf{a^{(t)}}$ and is often called the "total variation distance" between the distributions. We will build tools to upper bound $|\mathbf{p} - \mathbf{a^{(t)}}|_1$. Since $\mathbf{p}$ is the stationary distribution, the t for which $|\mathbf{p} - \mathbf{a^{(t)}}|_1$ becomes small is determined by the rate of convergence of the Markov chain to its steady state.

The following proposition is often useful.

Proposition 5.11 *For two probability distributions* $\mathbf{p}$ *and* $\mathbf{q}$*,*

$$|\mathbf{p} - \mathbf{q}|_1 = 2 \sum_i (p_i - q_i)^+ = 2 \sum_i (q_i - p_i)^+$$

where $x^+ = x$ *if* $x \geq 0$ *and* $x^+ = 0$ *if* $x < 0$.

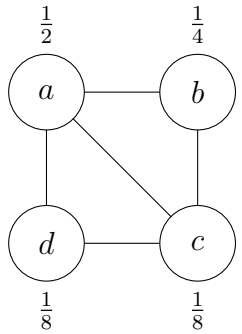
The proof is left as an exercise.

5.6.1 Metropolis-Hasting Algorithm

The Metropolis-Hasting algorithm is a general method to design a Markov chain whose stationary distribution is a given target distribution p. Start with a connected undirected graph G on the set of states. If the states are the lattice points $(x_1, x_2, \ldots, x_d)$ in $\mathbf{R}^d$ with $x_i \in \{0, 1, 2, , \ldots, n\}$, then G is the lattice graph with $2d$ coordinate edges at each interior vertex. In general, let r be the maximum degree of any vertex of G. The transitions of the Markov chain are defined as follows. At state i select neighbor j with probability $\frac{1}{r}$. Since the degree of i may be less than r, with some probability no edge is selected and the walk remains at i. If a neighbor j is selected and $p_j \geq p_i$, go to j. If $p_j < p_i$, go to j with probability p_j/p_i and stay at i with probability $1 - \frac{p_j}{p_i}$. Intuitively, this favors "heavier" states with higher p values. So, for $i \neq j$, adjacent in G,

$$p_{ij} = \frac{1}{r} \min \left(1, \frac{p_j}{p_i} \right)$$

$p(a) = \frac{1}{2}$
$p(b) = \frac{1}{4}$
$p(c) = \frac{1}{8}$
$p(d) = \frac{1}{8}$



$a \to b$	$\frac{1}{3}\ \frac{1}{4}\ \frac{2}{1} = \frac{1}{6}$	$c \to a$	$\frac{1}{3}$
$a \to c$	$\frac{1}{3}\ \frac{1}{8}\ \frac{2}{1} = \frac{1}{12}$	$c \to b$	$\frac{1}{3}$
$a \to d$	$\frac{1}{3}\ \frac{1}{8}\ \frac{2}{1} = \frac{1}{12}$	$c \to d$	$\frac{1}{3}$
$a \to a$	$1-\frac{1}{6}-\frac{1}{12}-\frac{1}{12}=\frac{2}{3}$	$c \to c$	$1-\frac{1}{3}-\frac{1}{3}-\frac{1}{3} = 0$
$b \to a$	$\frac{1}{3}$	$d \to a$	$\frac{1}{3}$
$b \to c$	$\frac{1}{3}\ \frac{1}{8}\ \frac{4}{1} = \frac{1}{6}$	$d \to c$	$\frac{1}{3}$
$b \to b$	$1-\frac{1}{3}-\frac{1}{6} = \frac{1}{2}$	$d \to d$	$1-\frac{1}{3}-\frac{1}{3} = \frac{1}{3}$

$$p(a) = p(a)p(a \to a) + p(b)p(b \to a) + p(c)p(c \to a) + p(d)p(d \to a)$$
$$= \tfrac{1}{2}\ \tfrac{2}{3} + \tfrac{1}{4}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{2}$$

$$p(b) = p(a)p(a \to b) + p(b)p(b \to b) + p(c)p(c \to b)$$
$$= \tfrac{1}{2}\ \tfrac{1}{6} + \tfrac{1}{4}\ \tfrac{1}{2} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{4}$$

$$p(c) = p(a)p(a \to c) + p(b)p(b \to c) + p(c)p(c \to c) + p(d)p(d \to c)$$
$$= \tfrac{1}{2}\ \tfrac{1}{12} + \tfrac{1}{4}\ \tfrac{1}{6} + \tfrac{1}{8}\ 0 + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{8}$$

$$p(d) = p(a)p(a \to d) + p(c)p(c \to d) + p(d)p(d \to d)$$
$$= \tfrac{1}{2}\ \tfrac{1}{12} + \tfrac{1}{8}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{8}$$

Figure 5.9: Using the Metropolis-Hasting algorithm to set probabilities for a random walk so that the stationary probability will be the desired probability.

and

$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}.$$

Thus,

$$p_i p_{ij} = \frac{p_i}{r} \min\left(1, \frac{p_j}{p_i}\right) = \frac{1}{r} \min(p_i, p_j) = \frac{p_j}{r} \min\left(1, \frac{p_i}{p_j}\right) = p_j p_{ji}.$$

By Lemma 5.3, the stationary probabilities are indeed $p(\mathbf{x})$ as desired.

Example: Consider the graph in Figure 5.9. Using the Metropolis-Hasting algorithm, assign transition probabilities so that the stationary probability of a random walk is $p(a) = \frac{1}{2}$, $p(b) = \frac{1}{4}$, $p(c) = \frac{1}{8}$, and $p(d) = \frac{1}{8}$. The maximum degree of any vertex is three, so at a, the probability of taking the edge (a, b) is $\frac{1}{3}\frac{1}{4}\frac{2}{1}$ or $\frac{1}{6}$. The probability of taking the edge (a, c) is $\frac{1}{3}\frac{1}{8}\frac{2}{1}$ or $\frac{1}{12}$ and of taking the edge (a, d) is $\frac{1}{3}\frac{1}{8}\frac{2}{1}$ or $\frac{1}{12}$. Thus, the probability of staying at a is $\frac{2}{3}$. The probability of taking the edge from b to a is $\frac{1}{3}$. The probability of taking the edge from c to a is $\frac{1}{3}$ and the probability of taking the edge from d to a is $\frac{1}{3}$. Thus, the stationary probability of a is $\frac{1}{4}\frac{1}{3} + \frac{1}{8}\frac{1}{3} + \frac{1}{8}\frac{1}{3} + \frac{1}{2}\frac{2}{3} = \frac{1}{2}$, which is the desired probability. ∎

5.6.2 Gibbs Sampling

Gibbs sampling is another Markov Chain Monte Carlo method to sample from a multivariate probability distribution. Let $p(\mathbf{x})$ be the target distribution where $\mathbf{x} = (x_1, \ldots, x_d)$. Gibbs sampling consists of a random walk on an undirectd graph whose vertices correspond to the values of $\mathbf{x} = (x_1, \ldots, x_d)$ and in which there is an edge from $\mathbf{x}$ to $\mathbf{y}$ if $\mathbf{x}$ and $\mathbf{y}$ differ in only one coordinate. Thus, the underlying graph is like a d-dimensional lattice except that the vertices in the same coordinate line form a clique.

To generate samples of $\mathbf{x} = (x_1, \ldots, x_d)$ with a target distribution $p(\mathbf{x})$, the Gibbs sampling algorithm repeats the following steps. One of the variables x_i is chosen to be updated. Its new value is chosen based on the marginal probability of x_i with the other variables fixed. There are two commonly used schemes to determine which x_i to update. One scheme is to choose x_i randomly, the other is to choose x_i by sequentially scanning from x_1 to x_d.

Suppose that $\mathbf{x}$ and $\mathbf{y}$ are two states that differ in only one coordinate. Without loss of generality let that coordinate be the first. Then, in the scheme where a coordinate is randomly chosen to modify, the probability $p_{\mathbf{xy}}$ of going from $\mathbf{x}$ to $\mathbf{y}$ is

$$p_{\mathbf{xy}} = \frac{1}{d} p(y_1 | x_2, x_3, \ldots, x_d).$$

Simplify following The normalizing constant is $1/d$ since for a given value i the probability distribution of $p(y_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ sums to one, and thus summing i over the d-dimensions results in a value of d. Similarly,

$$\begin{aligned} p_{\mathbf{yx}} =& \frac{1}{d} p(x_1 | y_2, y_3, \ldots, y_d) \\ =& \frac{1}{d} p(x_1 | x_2, x_3, \ldots, x_d). \end{aligned}$$

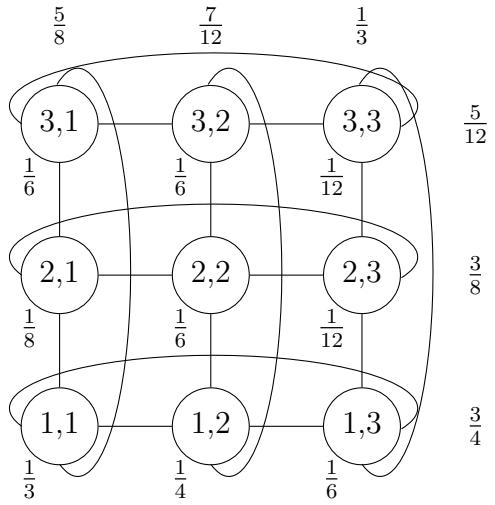
Here use was made of the fact that for $j \neq i$, $x_j = y_j$.

It is simple to see that this chain has stationary probability proportional to $p(\mathbf{x})$. Rewrite $p_{\mathbf{xy}}$ as

$$\begin{aligned} p_{\mathbf{xy}} &= \frac{1}{d} \frac{p(y_1 | x_2, x_3, \ldots, x_d) p(x_2, x_3, \ldots, x_d)}{p(x_2, x_3, \ldots, x_d)} \\ &= \frac{1}{d} \frac{p(y_1, x_2, x_3, \ldots, x_d)}{p(x_2, x_3, \ldots, x_d)} \\ &= \frac{1}{d} \frac{p(\mathbf{y})}{p(x_2, x_3, \ldots, x_d)} \end{aligned}$$

again using $x_j = y_j$ for $j \neq i$. Similarly write

$$p_{\mathbf{yx}} = \frac{1}{d} \frac{p(\mathbf{x})}{p(x_2, x_3, \ldots, x_d)}$$



$p(1,1) = \frac{1}{3}$
$p(1,2) = \frac{1}{4}$
$p(1,3) = \frac{1}{6}$
$p(2,1) = \frac{1}{8}$
$p(2,2) = \frac{1}{6}$
$p(2,3) = \frac{1}{12}$
$p(3,1) = \frac{1}{6}$
$p(3,2) = \frac{1}{6}$
$p(3,3) = \frac{1}{12}$

$p_{(11)(12)} = \frac{1}{d}p_{12}/(p_{11} + p_{12} + p_{13} = \frac{1}{2}\frac{1}{4}/(\frac{1}{3}\frac{1}{4}\frac{1}{6} = \frac{1}{2}\frac{1}{4}/\frac{9}{12} = \frac{1}{2}\frac{1}{4}\frac{4}{3} = \frac{1}{6}$
Calculation of edge probability $p_{(11)(12)}$

$p_{(11)(12)} = \frac{1}{2}\frac{1}{4}\frac{4}{3} = \frac{1}{6}$ $\quad p_{(12)(11)} = \frac{1}{2}\frac{1}{3}\frac{4}{3} = \frac{2}{9}$ $\quad p_{(13)(11)} = \frac{1}{2}\frac{1}{3}\frac{4}{3} = \frac{2}{9}$ $\quad p_{(21)(22)} = \frac{1}{2}\frac{1}{6}\frac{8}{3} = \frac{2}{9}$
$p_{(11)(13)} = \frac{1}{2}\frac{1}{6}\frac{4}{3} = \frac{1}{9}$ $\quad p_{(12)(13)} = \frac{1}{2}\frac{1}{6}\frac{4}{3} = \frac{1}{9}$ $\quad p_{(13)(12)} = \frac{1}{2}\frac{1}{4}\frac{4}{3} = \frac{1}{6}$ $\quad p_{(21)(23)} = \frac{1}{2}\frac{1}{12}\frac{8}{3} = \frac{1}{9}$
$p_{(11)(21)} = \frac{1}{2}\frac{1}{8}\frac{8}{5} = \frac{1}{10}$ $\quad p_{(12)(22)} = \frac{1}{2}\frac{1}{6}\frac{12}{7} = \frac{1}{7}$ $\quad p_{(13)(23)} = \frac{1}{2}\frac{1}{12}\frac{3}{1} = \frac{1}{8}$ $\quad p_{(21)(11)} = \frac{1}{2}\frac{1}{3}\frac{8}{5} = \frac{4}{15}$
$p_{(11)(31)} = \frac{1}{2}\frac{1}{6}\frac{8}{5} = \frac{2}{15}$ $\quad p_{(12)(32)} = \frac{1}{2}\frac{1}{6}\frac{12}{7} = \frac{1}{7}$ $\quad p_{(13)(33)} = \frac{1}{2}\frac{1}{12}\frac{3}{1} = \frac{1}{8}$ $\quad p_{(21)(31)} = \frac{1}{2}\frac{1}{6}\frac{8}{5} = \frac{2}{15}$
Edge probabilities.

$p_{11}p_{(11)(12)} = \frac{1}{3}\frac{1}{6} = \frac{1}{4}\frac{2}{9} = p_{12}p_{(12)(11)}$
$p_{11}p_{(11)(13)} = \frac{1}{3}\frac{1}{9} = \frac{1}{6}\frac{2}{9} = p_{13}p_{(13)(11)}$
$p_{11}p_{(11)(21)} = \frac{1}{3}\frac{1}{10} = \frac{1}{8}\frac{4}{15} = p_{21}p_{(21)(11)}$
Verification of a few edges.

Note that the edge probabilities out of a state such as (1,1) do not add up to one. That is, with some probability the walk stays at the state that it is in. For example, $p_{(11)(11)} = p_{(11)(12)} + p_{(11)(13)} + p_{(11)(21)} + p_{(11)(31)} = 1 - \frac{1}{6} - \frac{1}{24} - \frac{1}{32} - \frac{1}{24} = \frac{9}{32}$.

Figure 5.10: Using the Gibbs algorithm to set probabilities for a random walk so that the stationary probability will be a desired probability.

from which it follows that $p(\mathbf{x})p_{xy} = p(\mathbf{y})p_{yx}$. By Lemma 5.3 the stationary probability of the random walk is $p(\mathbf{x})$.

5.7 Areas and Volumes

Computing areas and volumes is a classical problem. For many regular figures in two and three dimensions there are closed form formulae. In Chapter 2, we saw how to compute volume of a high dimensional sphere by integration. For general convex sets in d-space, there are no closed form formulae. Can we estimate volumes of d-dimensional convex sets in time that grows as a polynomial function of d? The MCMC method answes this question in the affirmative.

One way to estimate the area of the region is to enclose it in a rectangle and estimate the ratio of the area of the region to the area of the rectangle by picking random points in the rectangle and seeing what proportion land in the region. Such methods fail in high dimensions. Even for a sphere in high dimension, a cube enclosing the sphere has exponentially larger area, so exponentially many samples are required to estimate the volume of the sphere.

It turns out that the problem of estimating volumes of sets is reducible to the problem of drawing uniform random samples from sets. Suppose one wants to estimate the volume of a convex set R. Create a concentric series of larger and larger spheres $S_1, S_2, \ldots, S_k$ such that S_1 is contained in R and S_k contains R. Then

$$\text{Vol}(R) = \text{Vol}(S_k \cap R) = \frac{\text{Vol}(S_k \cap R)}{\text{Vol}(S_{k-1} \cap R)} \frac{\text{Vol}(S_{k-1} \cap R)}{\text{Vol}(S_{k-2} \cap R)} \cdots \frac{\text{Vol}(S_2 \cap R)}{\text{Vol}(S_1 \cap R)} \text{Vol}(S_1)$$

If the radius of the sphere S_i is $1 + \frac{1}{d}$ times the radius of the sphere S_{i-1}, then the value of

$$\frac{\text{Vol}(S_{k-1} \cap R)}{\text{Vol}(S_{k-2} \cap R)}$$

can be estimated by rejection sampling provided one can select points at random from a d-dimensional region. Since the radii of the spheres grows as $1 + \frac{1}{d}$, the number of spheres is at most

$$O(\log_{1+(1/d)} R) = O(Rd).$$

It remains to show how to draw a uniform random sample from a d-dimensional set. It is at this point that we require the set to be convex so that the Markov chain technique we use will converge quickly to its stationary probability. To select a random sample from a d-dimensional convex set impose a grid on the region and do a random walk on the grid points. At each time, pick one of the $2d$ coordinate neighbors of the current grid point, each with probability $1/(2d)$ and go to the neighbor if it is still in the set; otherwise, stay put and repeat. If the grid length in each of the d coordinate directions is at most some a, the total number of grid points in the set is at most a^d. Although this is exponential in

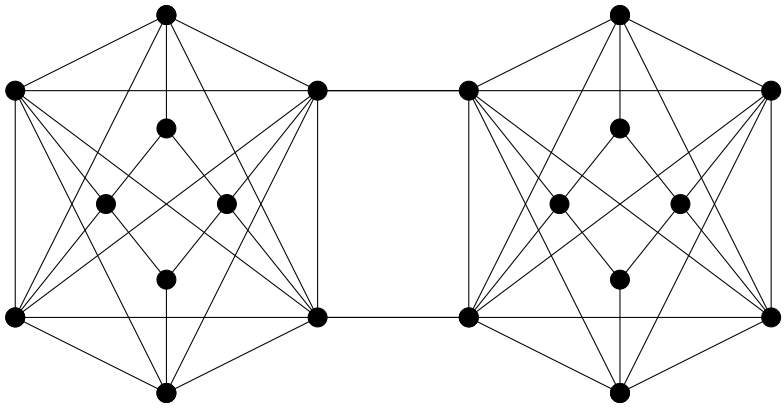


Figure 5.11: A network with a constriction.

d, the Markov chain turns out to be rapidly mixing (the proof is beyond our scope here) and leads to polynomial time bounded algorithm to estimate the volume of any convex set in $\mathbf{R}^d$.

5.8 Convergence of Random Walks on Undirected Graphs

The Metropolis-Hasting algorithm and Gibbs sampling both involve a random walk. Initial states of the walk are highly dependent on the start state of the walk. Both these walks are random walks on edge-weighted undirected graphs. Such Markov chains are derived from electrical networks. Recall the following notation which we will use throughout this section. Given a network of resistors, the conductance of edge (x, y) is denoted c_{xy} and the normalizing constant c_x equals $\sum_y c_{xy}$. The Markov chain has transition probabilities $p_{xy} = c_{xy}/c_x$. We assume the chain is connected. Since

$$c_x p_{xy} = c_c c_{xy}/c_x = c_{xy} = c_{yx} = c_y c_{yx}/c_y = c_y p_{xy}$$

the stationary probabilities are proportional to c_x where the normalization constant is $c_0 = \sum_x c_x$.

An important question is how fast the walk starts to reflect the stationary probability of the Markov process. If the convergence time was proportional to the number of states, the algorithms would not be very useful since the number of states can be exponentially large.

There are clear examples of connected chains that take a long time to converge. A chain with a constriction, see Figure 5.11, takes a long time to converge since the walk is unlikely to reach the narrow passage between the two halves, both of which are reasonably big. We will show in Theorem 5.12 that the time to converge is quantitatively related to

the tightest constriction.

A function is unimodal if it has a single maximum, i.e., it increases and then decreases. A unimodal function like the normal density has no constriction blocking a random walk from getting out of a large set of states, whereas a bimodal function can have a constriction. Interestingly, many common multivariate distributions as well as univariate probability distributions like the normal and exponential are unimodal and sampling according to these distributions can be done using the methods here.

A natural problem is estimating the probability of a convex region in d-space according to a normal distribution. One technique to do this is rejection sampling. Let R be the region defined by the inequality $x_1 + x_2 + \cdots + x_{d/2} \leq x_{d/2+1} + \cdots + x_d$. Pick a sample according to the normal distribution and accept the sample if it satisfies the inequality. If not, reject the sample and retry until one gets a number of samples satisfying the inequality. The probability of the region is approximated by the fraction of the samples that satisfied the inequality. However, suppose R was the region $x_1 + x_2 + \cdots + x_{d-1} \leq x_d$. The probability of this region is exponentially small in d and so rejection sampling runs into the problem that we need to pick exponentially many samples before we accept even one sample. This second situation is typical. Imagine computing the probability of failure of a system. The object of design is to make the system reliable, so the failure probability is likely to be very low and rejection sampling will take a long time to estimate the failure probability.

In general, there could be constrictions that prevent rapid convergence of a Markov chain to its stationary probability. However, if the set is convex in any number of dimensions, then there are no constrictions and there is rapid convergence although the proof of this is beyond the scope of this book.

We define below a combinatorial measure of constriction for a Markov chain, called the *normalized conductance*, and relate this quantity to the rate at which the chain converges to the stationarity probability. The conductance of an edge (x, y) leaving a set of states S is defined to be $\pi_x c_{xy}$ where π_x is the stationary probability of vertex x. One way to avoid constrictions like the one in the picture of Figure 5.11 is to insure that the total conductance of edges leaving every subset of states to be high. This is not possible if S was itself small or even empty. So, in what follows, we "normalize" the total conductance of edges leaving S by the size of S as measured by total c_x for $x \in S$. Recall that $p_{xy} = \frac{c_{xy}}{c_x}$ and the stationary probability $\pi_x = \frac{c_x}{c_0}$ where $c_0 = \sum_x c_x$. In defining the conductance of edges leaving a set we have ignored the normalizing constants.

Definition 5.1 *For a subset S of vertices, the normalized conductance $\Phi(S)$ of S is the*

ratio of the total conductance of all edges from S to $\bar{S}$ to the total of the c_x for $x \in S$.

$$\Phi(S) = \frac{\sum\limits_{(x,y)} c_{xy}}{\sum\limits_{x\in S} c_x} = \frac{\sum\limits_{(x,y)} c_x p_{xy}}{\sum\limits_{x\in S} c_0\pi_x} = \frac{\sum\limits_{(x,y)} c_0\pi_x p_{xy}}{\sum\limits_{x\in S} c_0\pi_x} = \frac{\sum\limits_{(x,y)} \pi_x p_{xy}}{\sum\limits_{x\in S} \pi_x}$$

∎

The normalized conductance[5] of S is the probability of taking a step from S to outside S conditioned on starting in S in the stationary probability distribution π. The stationary distribution for state x conditioned on being in S is

$$\frac{\pi_x}{\pi(S)} = \frac{c_x}{\sum\limits_{x\in S} c_x}.$$

where $\pi(S) = \sum\limits_{x\in S} \pi_x$.

Definition 5.2 *The* normalized conductance *of the Markov chain, denoted* Φ, *is defined by*

$$\Phi = \min_{\substack{S \\ \pi(S)\le 1/2}} \Phi(S).$$

∎

The restriction to sets with $\pi \le 1/2$ in the definition of Φ is natural. The definition of Φ guarantees that if Φ is high, there is high probability of moving from S to $\bar{S}$ so it is unlikely to get stuck in S provided $\pi(S) \le \frac{1}{2}$. If $\pi(S) > \frac{1}{2}$, say $\pi(S) = \frac{3}{4}$, then since for every edge $\pi_i p_{ij} = \pi_j p_{ji}$

$$\Phi(S) = \frac{\sum_{i\in S} \pi_i p_{ij}}{\sum_{i\in S} \pi_i} = \frac{\sum_{j\in \bar{S}} \pi_j p_{ji}}{3\sum_{j\in \bar{S}} \pi_k} = \Phi(\bar{S})/3$$

Since $\Phi(\bar{S}) \ge \Phi$, we still have at least $\Phi/3$ probability of moving out of S. The larger $\pi(S)$ is the smaller the probability of moving out, which is as it should be. We cannot move out of the whole set! One does not need to escape from big sets. Note that a constriction would mean a small Φ.

Definition 5.3 *Fix $\varepsilon > 0$. The* ε-mixing time *of a Markov chain is the minimum integer t such that for any starting distribution $\mathbf{p^{(0)}}$, the 1-norm distance between the t-step running average probability distribution*[6] *and the stationary distribution is at most ε.* ∎

[5]We will often drop the word "normalized" and just say "conductance".

[6]Recall that $\mathbf{a^{(t)}} = \frac{1}{t}(\mathbf{p^{(0)}} + \mathbf{p^{(1)}} + \cdots + \mathbf{p^{(t-1)}})$ is called the running average distribution.

The theorem below states that if Φ, the normalized conductance of the Markov chain, is large, then there is fast convergence of the running average probability. Intuitively, if Φ is large, the walk rapidly leaves any subset of states. Later we will see examples where the mixing time is much smaller than the cover time. That is, the number of steps before a random walk reaches a random state independent of its starting state is much smaller than the average number of steps needed to reach every state. In fact for some graphs, called expenders, the mixing time is logarithmic in the number of states.

Theorem 5.12 *The ε-mixing time of a random walk on an undirected graph is*

$$O\left(\frac{\ln(1/\pi_{min})}{\Phi^2\varepsilon^3}\right)$$

where π_{min} is the minimum stationary probability of any state.

Proof: Let

$$t = \frac{c\ln(1/\pi_{\min})}{\Phi^2\varepsilon^2},$$

for a suitable constant c. Let $\mathbf{a} = \mathbf{a^{(t)}}$ be the running average distribution for this value of t. We need to show that $|\mathbf{a} - \boldsymbol{\pi}| \leq \varepsilon$.

Let v_i denote the ratio of the long term average probability for state i at time t divided by the stationary probability for state i. Thus, $v_i = \frac{a_i}{\pi_i}$. Renumber states so that $v_1 \geq v_2 \geq \cdots$. A state i for which $v_i > 1$ has more probability than its stationary probability. Execute one step of the Markov chain starting at probabilities $\mathbf{a}$. The probability vector after that step is $\mathbf{a}P$. Now, $\mathbf{a} - \mathbf{a}P$ is the net loss of probability for each state due to the step. Let k be any integer with $v_k > 1$. Let $A = \{1, 2, \ldots, k\}$. A is a "heavy" set, consisting of states with $a_i \geq \pi_i$. The net loss of probability for each state from the set A in one step is $\sum_{i=1}^{k}(a_i - (\mathbf{a}P)_i) \leq \frac{2}{t}$ as in the proof of Theorem 5.2.

Another way to reckon the net loss of probability from A is to take the difference of the probability flow from A to $\bar{A}$ and the flow from $\bar{A}$ to A. For $i < j$,

$$\text{net-flow}(i,j) = \text{flow}(i,j) - \text{flow}(j,i) = \pi_i p_{ij} v_i - \pi_j p_{ji} v_j = \pi_j p_{ji}(v_i - v_j) \geq 0,$$

Thus, for any $l \geq k$, the flow from A to $\{k+1, k+2, \ldots, l\}$ minus the flow from $\{k+1, k+2, \ldots, l\}$ to A is nonnegative. At each step, heavy sets loose probability. Since for $i \leq k$ and $j > l$, we have $v_i \geq v_k$ and $v_j \leq v_{l+1}$, the net loss from A is at least

$$\sum_{\substack{i\leq k\\ j>l}} \pi_j p_{ji}(v_i - v_j) \geq (v_k - v_{l+1}) \sum_{\substack{i\leq k\\ j>l}} \pi_j p_{ji}.$$

Thus,

$$(v_k - v_{l+1}) \sum_{\substack{i\leq k\\ j>l}} \pi_j p_{ji} \leq \frac{2}{t}.$$

If the total stationary probability $\pi(\{i|v_i \leq 1\})$ of those states where the current probability is less than their stationary probability is less than $\varepsilon/2$, then

$$|\mathbf{a} - \boldsymbol{\pi}|_1 = 2 \sum_{\substack{i \\ v_i \leq 1}} (1 - v_i)\pi_i \leq \varepsilon,$$

so we are done. Assume $\pi(\{i|v_i \leq 1\}) > \varepsilon/2$ so that $\pi(A) \geq \varepsilon \min(\pi(A), \pi(\bar{A}))/2$. Choose l to be the largest integer greater than or equal to k so that

$$\sum_{j=k+1}^{l} \pi_j \leq \varepsilon \Phi \pi(A)/2.$$

Since

$$\sum_{i=1}^{k} \sum_{j=k+1}^{l} \pi_j p_{ji} \leq \sum_{j=k+1}^{l} \pi_j \leq \varepsilon \Phi \pi(A)/2$$

by the definition of Φ,

$$\sum_{i \leq k < j} \pi_j p_{ji} \geq \Phi \min(\pi(A), \pi(\bar{A})) \geq \varepsilon \Phi \pi(A).$$

Thus, $\sum_{\substack{i \leq k \\ j > l}} \pi_j p_{ji} \geq \varepsilon \Phi \pi(A)/2$. Substituting into the inequality 5.8 gives

$$v_k - v_{l+1} \leq \frac{8}{t \varepsilon \Phi \pi(A)}. \tag{5.5}$$

This inequality says that v does not drop too much as we go from k to $l+1$. On the other hand, the cumulative total of π will have increased, since, $\pi_1 + \pi_2 + \cdots + \pi_{l+1} \geq \rho(\pi_1 + \pi_2 + \cdots + \pi_k)$, where, $\rho = 1 + \frac{\varepsilon \Phi}{2}$. We will be able to use this repeatedly to argue that overall v does not drop too much. If that is the case (in the extreme, for example, if all the v_i are 1 each), then intuitively, $\mathbf{a} \approx \boldsymbol{\pi}$, which is what we are trying to prove. Unfortunately, the technical execution of this argument is a bit messy - we have to divide $\{1, 2, \ldots, n\}$ into groups and consider the drop in v as we move from one group to the next and then add up. We do this now.

Now, divide $\{1, 2, \ldots\}$ into groups as follows. The first group G_1 is $\{1\}$. In general, if the r^{th} group G_r begins with state k, the next group G_{r+1} begins with state $l+1$ where l is as defined above. Let i_0 be the largest integer with $v_{i_0} > 1$. Stop with G_m, if G_{m+1} would begin with an $i > i_0$. If group G_r begins in i, define $u_r = v_i$.

$$|\mathbf{a} - \boldsymbol{\pi}|_1 \leq 2 \sum_{i=1}^{i_0} \pi_i (v_i - 1) \leq \sum_{r=1}^{m} \pi(G_r)(u_r - 1) = \sum_{r=1}^{m} \pi(G_1 \cup G_2 \cup \ldots \cup G_r)(u_r - u_{r+1}),$$

where the analog of integration by parts for sums is used in the last step using the convention that $u_{m+1} = 1$. Since $u_r - u_{r+1} \leq 8/\varepsilon\Phi\pi(G_1 \cup \ldots \cup G_r)$, the sum is at most $8m/t\varepsilon\Phi$. Since $\pi_1 + \pi_2 + \cdots + \pi_{l+1} \geq \rho(\pi_1 + \pi_2 + \cdots + \pi_k)$,

$$m \leq \ln_\rho(1/\pi_1) \leq \ln(1/\pi_1)/(\rho - 1).$$

Thus $|\mathbf{a} - \boldsymbol{\pi}|_1 \leq O(\ln(1/\pi_{\min})/t\Phi^2\varepsilon^2) \leq \varepsilon$ for a suitable choice of c and this completes the proof. ∎

5.8.1 Using Normalized Conductance to Prove Convergence

We now apply Theorem 5.12 to some examples to illustrate how the normalized conductance bounds the rate of convergence. Our first examples will be simple graphs. The graphs do not have rapid converge, but their simplicity helps illustrate how to bound the normalized conductance and hence the rate of convergence.

A 1-dimensional lattice

Consider a random walk on an undirected graph consisting of an n-vertex path with self-loops at the both ends. With the self loops, the stationary probability is a uniform $\frac{1}{n}$ over all vertices. The set with minimum normalized conductance is the set with the maximum number of vertices with the minimum number of edges leaving it. This set consists of the first $n/2$ vertices, for which total conductance of edges from S to $\bar{S}$ is $\pi_{n/2}p_{n/2,\frac{n}{2}+1} = \Omega(\frac{1}{n})$ and $\pi(S) = \frac{1}{2}$. ($\pi_{\frac{n}{2}}$ is the stationary probability of vertex numbered $\frac{n}{2}$.) Thus

$$\Phi(S) = 2\pi_{\frac{n}{2}}\, p_{\frac{n}{2},\frac{n}{2}+1} = \Omega(1/n).$$

By Theorem 5.12, for ε a constant such as 1/100, after $O(n^2 \log n)$ steps, $|\mathbf{a}^{(\mathbf{t})} - \boldsymbol{\pi}|_1 \leq 1/100$. This graph does not have rapid convergence. The hitting time and the cover time are $O(n^2)$. In many interesting cases, the mixing time may be much smaller than the cover time. We will see such an example later.

A 2-dimensional lattice

Consider the $n \times n$ lattice in the plane where from each point there is a transition to each of the coordinate neighbors with probability 1/4. At the boundary there are self-loops with probability 1-(number of neighbors)/4. It is easy to see that the chain is connected. Since $p_{ij} = p_{ji}$, the function $f_i = 1/n^2$ satisfies $f_i p_{ij} = f_j p_{ji}$ and by Lemma 5.3 is the stationary probability. Consider any subset S consisting of at most half the states. Index states by their x and y coordinates. For at least half the states in S, either row x or column y intersects $\bar{S}$ (Exercise 5.46). So at least $\Omega(|S|/n)$ points in S are adjacent to points in $\bar{S}$. Each such point contributes $\pi_i p_{ij} = \Omega(1/n^2)$ to flow$(S, \bar{S})$. So

$$\sum_{i \in S} \sum_{j \in \bar{S}} \pi_i p_{ij} \geq c|S|/n^3.$$

Thus, $\Phi \geq \Omega(1/n)$. By Theorem 5.12, after $O(n^2 \ln n/\varepsilon^2)$ steps, $|\mathbf{a}^{(\mathbf{t})} - \boldsymbol{\pi}|_1 \leq 1/100$.

A lattice in d-dimensions

Next consider the $n \times n \times \cdots \times n$ lattice in d-dimensions with a self-loop at each boundary point with probability $1 -$ (number of neighbors)$/2d$. The self loops make all π_i equal to n^{-d}. View the lattice as an undirected graph and consider the random walk on this undirected graph. Since there are n^d states, the cover time is at least n^d and thus exponentially dependent on d. It is possible to show (Exercise 5.62) that Φ is $\Omega(1/dn)$. Since all π_i are equal to n^{-d}, the mixing time is $O(d^3 n^2 \ln n/\varepsilon^2)$, which is polynomially bounded in n and d.
The d-dimensional lattice is related to the Metropolis-Hastings algorithm and Gibbs sampling although in those constructions there is a nonuniform probability distribution at the vertices. However, the d-dimension lattice case suggests why the Metropolis-Hastings and Gibbs sampling constructions might converge fast.

A clique

Consider an n vertex clique with a self loop at each vertex. For each edge, $c_{xy} = 1$ and thus for each vertex, $c_x = n$. Let S be a subset of the vertices. Then

$$\sum_{x \in S} c_x = n|S|.$$

$$\sum_{(x,y)} c_{xy} = |S||\overline{S}|$$

and

$$\Phi(S) = \frac{\sum_{(x,y)} c_{xy}}{\sum_{x \in S} c_x} = \frac{|\overline{S}|}{n}.$$

Now $\Phi = min\ \Phi(S)$ for $|S| \leq \frac{n}{2}$ and hence $|\overline{S}| \geq \frac{n}{2}$. Thus $\Phi = \frac{1}{2}$. This gives a mixing time of

$$O\left(\frac{\ln \frac{1}{\pi_{\min}}}{\Phi^2 \epsilon^3}\right) = O\left(\frac{\ln n}{\frac{1}{4}\epsilon^3}\right) = O(\ln n).$$

A connected undirected graph

Next consider a random walk on a connected n vertex undirected graph where at each vertex all edges are equally likely. The stationary probability of a vertex equals the degree of the vertex divided by the sum of degrees which equals twice the number of edges. The sum of the vertex degrees is at most n^2 and thus, the steady state probability of each vertex is at least $\frac{1}{n^2}$. Since the degree of a vertex is at most n, the probability of each edge

at a vertex is at least $\frac{1}{n}$. For any S, the total conductance of edges out of S is greater than or equal to

$$\frac{1}{n^2}\frac{1}{n} = \frac{1}{n^3}.$$

Thus, Φ is at least $\frac{1}{n^3}$. Since $\pi_{\min} \geq \frac{1}{n^2}$, $\ln \frac{1}{\pi_{min}} = O(\ln n)$. Thus, the mixing time is $O(n^6(\ln n)/\varepsilon^2)$.

The Gaussian distribution on the interval [-1,1]

Consider the interval $[-1, 1]$. Let δ be a "grid size" specified later and let G be the graph consisting of a path on the $\frac{2}{\delta} + 1$ vertices $\{-1, -1 + \delta, -1 + 2\delta, \ldots, 1 - \delta, 1\}$ having self loops at the two ends. Let $\pi_x = ce^{-\alpha x^2}$ for $x \in \{-1, -1 + \delta, -1 + 2\delta, \ldots, 1 - \delta, 1\}$ where $\alpha > 1$ and c has been adjusted so that $\sum_x \pi_x = 1$.

We now describe a simple Markov chain with the π_x as its stationary probability and argue its fast convergence. With the Metropolis-Hastings' construction, the transition probabilities are

$$p_{x,x+\delta} = \frac{1}{2}\min\left(1, \frac{e^{-\alpha(x+\delta)^2}}{e^{-\alpha x^2}}\right) \text{ and } p_{x,x-\delta} = \frac{1}{2}\min\left(1, \frac{e^{-\alpha(x-\delta)^2}}{e^{-\alpha x^2}}\right).$$

Let S be any subset of states with $\boldsymbol{\pi}(S) \leq \frac{1}{2}$. Consider the case when S is an interval $[k\delta, 1]$ for $k \geq 1$. It is easy to see that

$$\begin{aligned}
\boldsymbol{\pi}(S) &\leq \int_{x=(k-1)\delta}^{\infty} ce^{-\alpha x^2}\, dx \\
&\leq \int_{(k-1)\delta}^{\infty} \frac{x}{(k-1)\delta} ce^{-\alpha x^2}\, dx \\
&= O\left(\frac{ce^{-\alpha((k-1)\delta)^2}}{\alpha(k-1)\delta}\right).
\end{aligned}$$

Now there is only one edge from S to $\bar{S}$ and total conductance of edges out of S is

$$\sum_{i\in S}\sum_{j\notin S} \pi_i p_{ij} = \pi_{k\delta} p_{k\delta,(k-1)\delta} = \min(ce^{-\alpha k^2\delta^2}, ce^{-\alpha(k-1)^2\delta^2}) = ce^{-\alpha k^2\delta^2}.$$

Using $1 \leq k \leq 1/\delta$ and $\alpha \geq 1$, $\Phi(S)$ is

$$\begin{aligned}
\Phi(S) = \frac{\text{flow}(S, \bar{S})}{\pi(S)} &\geq ce^{-\alpha k^2\delta^2} \frac{\alpha(k-1)\delta}{ce^{-\alpha((k-1)\delta)^2}} \\
&\geq \Omega(\alpha(k-1)\delta e^{-\alpha\delta^2(2k-1)}) \geq \Omega(\delta e^{-O(\alpha\delta)}).
\end{aligned}$$

For $\delta < \frac{1}{\alpha}$, we have $\alpha\delta < 1$, so $e^{-O(\alpha\delta)} = \Omega(1)$, thus, $\Phi(S) \geq \Omega(\delta)$. Now, $\pi_{\min} \geq ce^{-\alpha} \geq$

$e^{-1/\delta}$, so $\ln(1/\pi_{\min}) \leq 1/\delta$.

If S is not an interval of the form $[k, 1]$ or $[-1, k]$, then the situation is only better since there is more than one "boundary" point which contributes to flow$(S, \bar{S})$. We do not present this argument here. By Theorem 5.12 in $\Omega(1/\delta^3\varepsilon^2)$ steps, a walk gets within ε of the steady state distribution.

In these examples, we have chosen simple probability distributions. The methods extend to more complex situations.

5.9 Bibliographic Notes

The material on the analogy between random walks on undirected graphs and electrical networks is from [DS84] as is the material on random walks in Euclidean space. Additional material on Markov chains can be found in [MR95b], [MU05], and [per10]. For material on Markov Chain Monte Carlo methods see [Jer98] and [Liu01].

The use of normalized conductance to prove convergence of Markov Chains is by Sinclair and Jerrum, [SJ] and Alon [Alo86]. A polynomial time bounded Markov chain based method for estimating the volume of convex sets was developed by Dyer, Frieze and Kannan [DFK91].